AD-A229 711

Technical Document 1930
September 1990

Specification for a Surface-Search Radar-Detection-Range Model

Claude P. Hattan

Approved for public release; distribution is unlimited.

NAVAL OCEAN SYSTEMS CENTER
San Diego, California 92152-5000

J. D. FONTANA, CAPT, USN
Commander

R. M. HILLYER
Technical Director

ADMINISTRATIVE INFORMATION

The work described in this document was performed from January 1990 to June 1990 by the Tropospheric Branch, Code 543, Naval Ocean Systems Center (NOSC), for the Naval Oceanographic Office, Stennis Space Center, Bay St. Louis, MS.

Released by
H. V. Hitney, Head
Tropospheric Branch

Under authority of
J. H. Richter, Head
Ocean and Atmospheric
Sciences Division

CONTENTS

FIGURES

TABLES

1.0 INTRODUCTION

A surface-search radar-detection-range model has been developed at the Naval Ocean Systems Center (NOSC). The model can be used to assess the effect of the environment on the performance of surface-search radar systems. Additionally, the effects of radar cross section (RCS) variability as a function of viewing angle, ship displacement, height of the ship, and range are also modeled. Two simplified ship target RCS models are combined to calculate the RCS variability. One of the simplified models provides RCS as a function of viewing angle, while the other gives the distribution of RCS as a function of height of the target vessel. Five classes of ship targets are modeled, ranging from small (patrol boats) to very large warships (aircraft carriers). Detection ranges for other ship classes can be inferred from their size relative to the ship targets that are used in the model. Environmental effects are accounted for by incorporating the U.S. Navy Oceanographic and Atmospheric Library (OAML) Standard Electromagnetic (EM) Propagation Model (Hattan, 1990) into the model. The software implementation of the model is written in ANSI Fortran 77, with MIL-STD-1753 extensions. The program provides the user with a table of expected detection ranges for the various target ship classes when the program is supplied with the proper radar system and environmental inputs.

2.0 INPUTS, OUTPUTS, AND LIMITS

2.1 INPUTS

A number of radar system and environmental inputs are required to determine the detection range of a surface-search radar system. The necessary radar system parameters are given in Table 1. The required environmental inputs are provided in Table 2. The antenna beamwidth and elevation angle parameters of Table 1 are not required for an omnidirectional antenna type. The input parameters of Table 1 and 2 are identical to the required inputs of the OAML Standard EM Propagation Model, with three exceptions. The receiver/target height entry of the Standard EM Model is not used and two new inputs are required: the free-space detection range of the radar for a 1 m^2 target and the maximum instrumented (unambiguous) range of the radar. The radar free-space range can be calculated by using the OAML Radar Free-space Detection Range Model (Hattan, 1989) and the unambiguous range obtained from either the radar system manual or calculated by using the method described by Patterson (1988). Shorthand variable names are given for inputs that are used in equations or other tables in subsequent sections of the text.

Table 1. Required surface-search radar system inputs.

Parameter	Units	Valid Input Range
Frequency, f	MHz	100.0 to 20,000.0
Height of Transmitting Antenna, H_t	m	1.0 to 100.0
Transmitting Antenna Polarization	n/a	Horizontal, vertical, or circular
Transmitting Antenna Type	n/a	Omnidirectional, $\sin(x)/x$, cosecant-squared, height-finder, or specific-system height-finder
Antenna Beamwidth, BW	deg	>0 to 45.0
Antenna Elevation Angle, μ_o	deg	-10.0 to 10.0
Maximum Instrumented Range, R_i	nmi	10 to 200.0
Radar Free-Space Range for 1 m^2 Target, R_1	nmi	1 to 1000.0

Table 2. Required environmental inputs.

Parameter	Units	Valid Input Range
Evaporation Duct Height, δ	m	0.0 to 40.0
Surface Wind Speed, W_s	kt	0.0 to 50.0
Height Array, H_i— 2 to 30 Elements	m	0.0 to 10,000.0
M-unit Array, M_i— Each Element Corresponding to the Like-Number Height Array Element	M	0.0 to 2000.0

2.2 OUTPUTS

The only output is a table of predicted detection ranges for five ship target classes for the specified inputs of Tables 1 and 2. The target ship classes are (1) aircraft carrier, (2) cruiser, (3) destroyer, (4) frigate, and (5) patrol boat. Sample program outputs for a variety of environmental and radar system inputs are presented in section 4.0.

2.3 LIMITS

The surface-search detection-range model described in this document will return a range value in nautical miles, limited to 200 nmi, for radar system operational parameters within the range of validity of the inputs of Table 1 and for environmental inputs within the range of validity of Table 2.

3.0 SURFACE-SEARCH RADAR-DETECTION-RANGE MODEL

The detection of large, complex targets by a radar is difficult to model accurately. Small targets, such as small aircraft or missiles, in the far field of a radar system are usually assumed to intercept and scatter a plane wave, which is equivalent to saying that they behave as point source targets. This is generally not the case for ship targets. Ships are large and complex enough to scatter radiation that is not a plane wave, especially within the radar horizon. Large fluctuations in received signal strength as a function of range, aspect angle, target size, radar antenna height, and the propagation environment can occur, and these signal excursions are equivalent to variations in the RCS of the target vessel. Any analysis of the variations of ship target RCS generally requires that a simplified model of target shape be used. The following subsections describe two RCS models which can be combined to predict detection ranges for various classes of ship targets. One model describes the variation of RCS as a function of aspect angle. This model can be used to determine the minimum, average, and maximum detection range of a ship target by establishing the radar receiver detection threshold corresponding to each of these RCS values. A second model, which gives the height variation of RCS, allows the propagation loss for any range to be determined.

3.1 RADAR CROSS SECTION OF SHIP TARGETS AS A FUNCTION OF ASPECT

The RCS of a ship is quite dependent on the azimuthal angle of incidence (aspect). Figure 1 is an example of a fairly typical variation of RCS with aspect as measured by Queen and Maine (1971). The polar diagram of Fig. 1 has three lines plotted which represent the 20, 50, and 80 percentile values of the RCS distribution function, inner-to-outermost curves, respectively. The RCS values are plotted in units of decibels above a 1 square-meter target (dBsm). There is an increase in RCS abeam and significant reductions in RCS at near-bow or near-stern angles of incidence, which should be expected. The fluctuations of RCS with aspect, of 10 dB or greater, can cause significant variations in detection range.

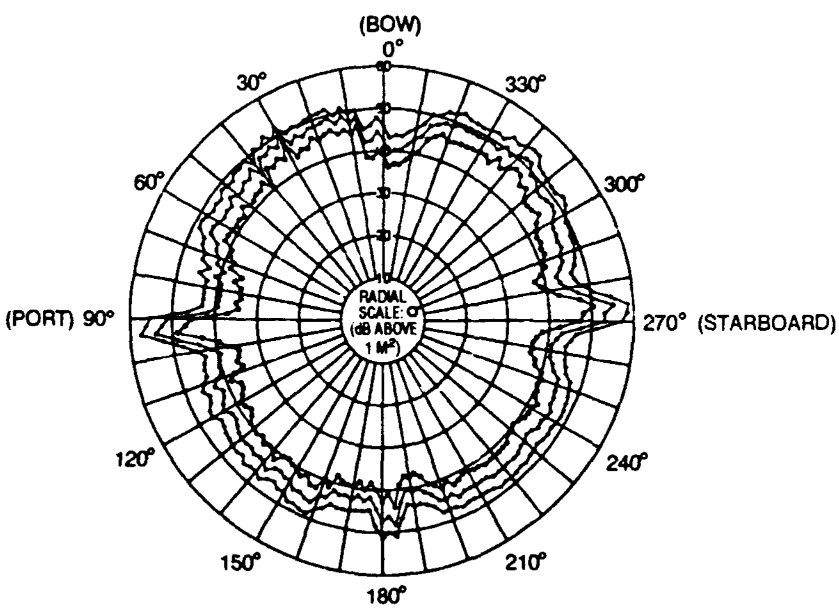

Figure 1. Typical ship RCS variation with aspect.

Skolnik (1974) has shown that the total radar cross section of a ship target can be related to the displacement of the vessel and the radar frequency. This relationship is given by

$$\sigma = 52 f^{1/2} D^{3/2} \tag{1}$$

where f is the radar frequency in megahertz and D is the ship's displacement in kilotons. This equation was developed based on an empirical fit to values of the measured 50th percentiles of the RCS distribution function (omitting the broadside peak) on data obtained by the Naval Research Laboratory's Target Characteristics Branch. Inspection of these data indicates approximately an 8-dB excursion from this average value to the measured RCS minimums. Similarly, a 13-dB range from average to the broadside peak characterizes the maximum RCS.

3.2 RADAR CROSS SECTION OF SHIP TARGETS AS A FUNCTION OF HEIGHT

The RCS variation shown in Fig. 1 is not solely due to the variation in total area of the ship as a function of aspect angle. There is also a considerable variation in RCS as a function of height. The hull, superstructure, and mast/antenna areas all have different microwave reflective properties because of their shapes and sizes. The fact that the various radar reflectors are not concentrated at a single range, or orientation, further complicates the nature of the reflected wave. The reflected energy as "seen" by the radar receiver is, of course, the sum of the signals returned from the various reflecting surfaces illuminated by the radar.

The basic transmission equation for a monostatic radar, from Kerr (1951), is

$$\frac{P_r}{P_t} = \frac{(G \lambda)^2 \sigma F^4}{(4\pi)^3 R^4} \tag{2}$$

where

- P_r is the power received
- P_t is the power transmitted
- G is the antenna gain
- λ is the wavelength
- σ is the RCS of the target
- F is the pattern propagation factor which, is defined as the ratio of the actual electric field at R to the free-space electric field at R

If there are n multiple targets at the same range, R, then the power received is the sum of the reflected energies from the various targets. If we assume that the cross sections of these n multiple targets at R are defined by $\sigma(h)$, where $\sigma(h)$ is the cross section of the target appropriate to height h, and all of the $\sigma(h)$'s are fully contained within the vertical and horizontal beamwidth of the radar, then Eq. 2 can be rewritten

$$\frac{P_r}{P_t} = \frac{(G \lambda)^2 \sum_n \sigma(h) \overline{F(h)}^4}{(4\pi)^3 R^4} \tag{3}$$

provided $F(h)$ does not vary significantly across the surface of $\sigma(h)$. If $F(h)$ does vary over the vertical extent of $\sigma(h)$, but $\sigma(h)$ is uniformly distributed across the same vertical extent, then the average F^4, defined as $\overline{F(h)}^4$, can be used in Eq. 3 to replace $F(h)^4$. If $\sigma(h)$ is replaced by $\sigma\, w(h)$, where

$w(h)$ represents the fractional weighting factor for the cross section represented by each $\sigma(h)$, then the summation on the right side of Eq. 3 can be replaced by

$$\sum_n \sigma(h)\ F(h)^4 = \sigma \sum_n w(h)\ \overline{F(h)}^4 \tag{4}$$

with the restriction that the summation over the $w(h) = 1$. This equation allows a ship target to be treated as if it were an ensemble of individual targets (which it is) by summing the individual contributions to the total ship RCS over the vertical extent of the ship, assuming that the variation in RCS is only a function of height.

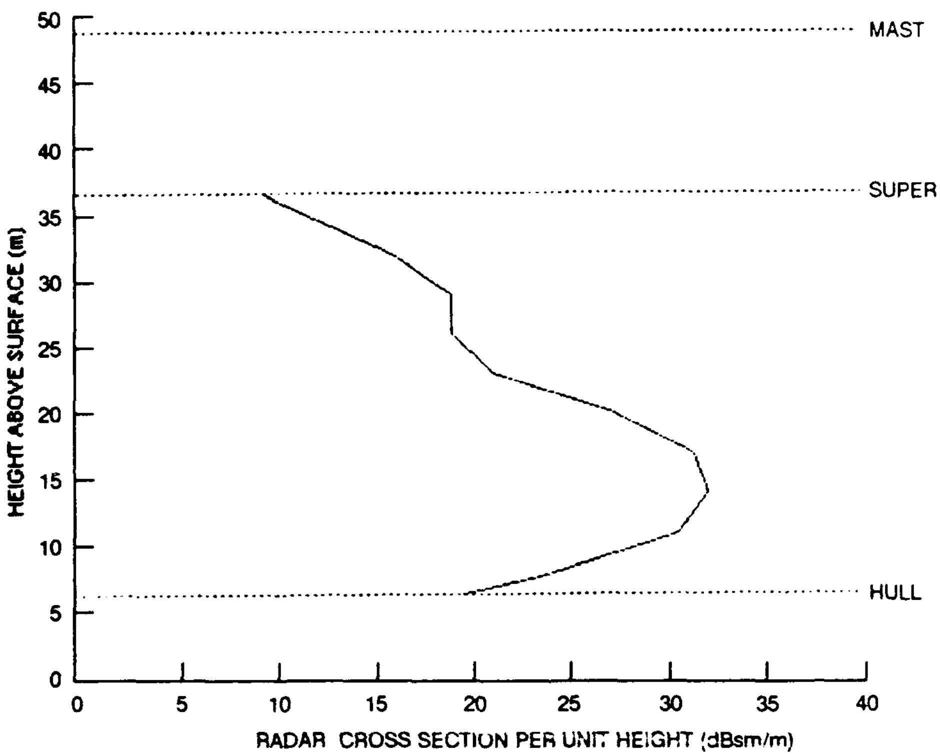

Figure 2. Ship radar cross section as a function of height.

Anderson (1986) has reported on a model developed at the NOSC by Hitney that gives the RCS distribution with height above the waterline. An example produced by this model is given in Fig. 2 for a cruiser-size target. RCS is expressed as decibels above a one square-meter target per meter of height (dBsm/m). The labels at the right side of Fig. 2 indicate the area of the ship responsible for the RCS. "Hull" is used to define the area from the waterline to the top of the main deck. "Super" is the height region above the main deck, including all major radar antennas, that comprises the superstructure of the ship. "Mast" applies to those heights above the superstructure, which primarily consist of mast and whip antenna structures. These boundaries are shown by the horizontal dashed lines. Note that in this model the entire RCS of the ship is due to return from the superstructure. In reality there is some return from both the hull and the mast areas, but it is primarily the superstructure, with its many reflectors, that is the main contributor to the total RCS of the ship. Anderson (1986) has compared ranges by using the above model and one that includes a small contribution from both mast and hull areas, and the ranges are nearly identical for a wide range of environments. The distribution of RCS per unit height for other ship targets is assumed identical to that shown in Fig. 2.

Table 3. Ship target RCS weighting factors.

Height (m)	Ship Class CV	CG	DD	FF	PB
2.50					0.0467
3.50					0.2134
4.50					0.3083
5.50			0.0151	0.0212	0.2625
6.50			0.0151	0.0212	0.1045
7.50		0.0151	0.0151	0.0818	0.0245
8.50		0.0151	0.0634	0.0970	0.0151
9 50		0.0151	0.0688	0.1229	0.0151
10.50		0.0634	0.0688	0.1402	0.0074
11.50		0.0688	0.0933	0.1318	0.0025
12.50		0.0688	0.0995	0.1193	
13.50		0.0933	0.0995	0.1049	
14.50		0.0995	0.0891	0.0475	
15.50		0.0995	0.0847	0.0475	
16.50		0.0891	0.0847	0.0111	
17.50	0.0180	0.0847	0.0541	0.0111	
18.50	0.0180	0.0847	0.0337	0.0077	
19.50	0.0436	0.0541	0.0337	0.0069	
20.50	0.0821	0.0337	0.0208	0.0069	
21.50	0.0821	0.0337	0.0079	0.0069	
22.50	0.1113	0.0208	0.0079	0.0055	
23.50	0.1186	0.0079	0.0067	0.0034	
24.50	0.1151	0.0079	0.0049	0.0029	
25.50	0.1010	0.0067	0.0049	0.0011	
26.50	0.1010	0.0049	0.0049	0.0011	
27.50	0.0645	0.0049	0.0049		
28.50	0.0402	0.0049	0.0049		
29.50	0.0402	0.0049	0.0044		
30.50	0.0094	0.0049	0.0024		
31.50	0.0094	0.0044	0.0024		
32.50	0.0080	0.0024	0.0022		
33.50	0.0058	0.0024	0.0008		
34.50	0.0058	0.0022	0.0008		
35.50	0.0058	0.0008	0.0008		
36.50	0.0058	0.0008			
37.50	0.0052	0.0008			
38.50	0.0028				
39.50	0.0028				
40.50	0.0017				
41.50	0.0010				
42.50	0.0010				

Using Hitney's RCS distribution model, it is possible to construct a table of weighting factors, w(h), for several different ship targets. Table 3 contains the weighting factors for five classes of ship targets and the height at which the factors were evaluated. The ship classes represented are a patrol boat, a frigate, a destroyer, a cruiser, and an aircraft carrier. A discrete height interval of 1 meter was used to construct the table, and the weighting factors were evaluated at the midpoint of the interval. Superstructure height measurements were obtained from *Jane's Fighting Ships* (1980) and represent a ship that is approximately the class average for US Navy vessels. The patrol boat is the exception, but it is representative of the Soviet 210-ton Stenka- and Osa-class boats, which are in world-wide operational use. The frigate measurements are taken from a 3400-ton Garcia-class ship, the destroyer's from the 5000-ton Farragut class, the cruiser's from the 10,110-ton Virginia class and the aircraft carrier's are from the 85,360-ton Kitty Hawk class.

3.3 CALCULATIONS OF DETECTION RANGE USING THE RADAR CROSS-SECTION MODELS

Using the distributed targets of Table 3, it is possible to use Eq. 1, 3, and 4 to calculate the total received power, P_r, at any range, R, if $F(h)$ is known. When P_r is equal to (or greater than) the radar receiver's signal-to-noise (detection) threshold, the target can be detected. The greatest range where this occurs corresponds to the maximum detection range for the average RCS value obtained from Eq. 1. $F(h)$ can be obtained as a function of range by using the OAML Standard EM Propagation Model. The Standard EM Propagation Model returns the pattern propagation factor, F, in decibels, for user-specified geometric, environmental and EM system inputs. F accounts for all of the propagation effects on the radar system. Assuming that the average value of $F(h)$ is constant over a reasonably small height interval of the target vessel, Eq. 4 can be used to sum the contributions from each individual height element by using the weighting factors obtained from Table 3. The average value of $F(h)$ is evaluated at the same height as the weighting factors of Table 3. By a proper selection of the range, R, the actual detection range for any ship target can be determined in an iterative manner.

The radar system detection threshold is determined from the free-space detection range input of Table 1. This input is based on the detection of a 1 square-meter target, but can be scaled to the actual size target by using the following relationship (Kerr, 1951):

$$R_{fs} = R_1 \sigma^{1/4} \tag{5}$$

where R_{fs} is the radar free space range for the desired target cross section, R_1 is the radar free-space range for a 1 square-meter target, and σ is the average target cross section from Eq. 1. R_{fs} can be used to determine the equivalent one-way propagation loss that the radar can sustain and still detect the target. The one-way propagation loss (detection threshold), in decibels, for the average RCS of the ship target is obtained by using the following equation (Kerr, 1951):

$$L_{avg} = 32.44 + 20\log(f) + 20\log(R_{fs}) \tag{6}$$

where f is the radar frequency in megahertz and R_{fs} is in kilometers. The actual one-way propagation loss, L, at any range can be determined by using the following equation (Kerr, 1951):

$$L = 32.44 + 20\log(f) + 20\log(R) - 20\log(F) \tag{7}$$

The greatest range where $L_{avg} = L$ is the actual detection range of the desired target. Substituting the summation of Eq. 4 into Eq. 7 yields

$$L = 32.44 + 20\log(f) + 20\log(R) - 5\log\left[\sum_n w(h)\ \overline{F(h)^4}\right] \tag{8}$$

Equation 8 is the most useful form of the loss equation to determine the detection range (i.e., where $L = L_{avg}$).

Equation 8 can be used to determine the minimum and maximum detection ranges as a function of aspect for ship targets also. Since the broadside maximum of the RCS distribution function is approximately 13 dB greater than the average RCS, the maximum detection threshold is given by

$$L_{\max} = L_{avg} + 13 \tag{9}$$

Similarly, the minimum RCS of the ship is 8 dB (at near-bow or near-stern angles) less than the average value, so that the detection threshold is

$$L_{\min} = L_{avg} - 8 \tag{10}$$

Since $L_{\max}$ is greater than L_{avg}, the corresponding detection range will also be greater, and it follows that L_{min} will yield a shorter detection range.

3.4 SURFACE-SEARCH DETECTION-RANGE-MODEL FORTRAN PROGRAM

The surface-search radar-detection-range model is implemented in a program called SSRDRT. SSRDRT is written in ANSI Fortran 77 with the allowable MIL-STD-1753 extensions. SSRDRT calculates three detection ranges for each of the five ship targets of Table 3 for the specified EM system and environmental parameters of Table 1 and Table 2. To use SSRDRT, the operator must compile and link the routines that constitute the program. A complete list of all subroutines is included in the appendix. The subroutines are listed in alphabetical order following lists of the MAIN and SSRDRT routines, the TARGET block data, and the common block "include" files. No EM system or environmental libraries are supplied with SSRDRT, though a limited number of environmental and EM system data sets for test purposes are listed in section 4.0.

A demonstration program, MAIN, which acts as a driver for the SSRDRT program is included to demonstrate the use of the program. The driver simply calls SSRDRT to initiate the program. In addition to MAIN, several subroutines allow the operator to enter environmental and EM system data from the keyboard or files. These subroutines, SYSFIL, ENVFIL, ENVINP, and SYSINP, are not intended as part of the SSRDRT program, but are only for use in verifying the correct operation of the SSRDRT program. Another routine, PRNRNG, is similarly included. PRNRNG prints a table of the range values, obtained from SSRDRT, for each of the five targets and the three detection thresholds for each target.

All but five of the subroutines that constitute SSRDRT are subroutines that are part of the OAML Standard EM Propagation Model, FFACTR. The operation of these subroutines is documented elsewhere (Hattan, 1990) and will not be explained in any great depth here. Since SSRDRT uses a single set of environmental and EM system inputs for determining the detection ranges of the five targets, FFACTR was modified by placing the ANTPAR, DCONST, GETK, MPROF, and OPCNST subroutine calls in SSRDRT. This was suggested by the FFACTR documentation to avoid redundant subroutine calls. The operational sequence of SSRDRT is detailed in the following paragraphs.

The MAIN program calls ENVINP and SYSINP to allow the operator to enter the environmental and radar parameters of Table 1 and Table 2. Subroutine TARGET contains the block data for the five ship targets' RCS weighting factors and heights of Table 3 and an array of the target ships' tonnage to the 3/2 power. Once these parameters have been entered, SSRDRT is called. SSRDRT initializes the FFACTR program constants by calling, in order, the MPROF, GETK, ANTPAR, OPCNST, and DCONST subroutines. After the constants have been initialized, subroutine RTLOOP is called to determine the detection ranges for each of the targets and thresholds. When the detection ranges have been calculated, RTLOOP terminates and returns to SSRDRT, which then returns to MAIN. MAIN then calls PRNRNG to print the range values. The program is then terminated.

RTLOOP is the subroutine which implements Eq. 8 to build the surface-search radar-detection-range table. The subroutine is essentially composed of two nested DO loops. The outer loop is the controlling loop for the five targets and the inner loop calculates the detection range for the three detection thresholds, L_{min}, L_{avg}, and L_{max}, as a function of target aspect angle.

RTLOOP calculates the threshold values, L_{min}, etc., for the individual ship targets. RTLOOP also calls subroutine RARRAY to build a three-element array for each target, which is used in a search scheme to minimize the time necessary to determine the detection range. The first element is 370 km (200 nmi), the maximum range allowed; the second is the minimum range at which the diffraction field calculations are valid, r_d; and the third is the range to the end of the optical interference region, r_o. r_o is determined by a call to the FFACTR subroutine OPLIMIT. r_d and r_o are computed by using the height closest to the median value of the target RCS obtained from Table 3 (15.5 meters for the cruiser). This height is very nearly 1/3 of the superstructure height difference above the main deck, as shown in Fig. 2. These ranges only approximate the overall behavior of the target, but are useful in shortening the search time. The one-way propagation loss of Eq. 8 is obtained by referencing the FLOOP subroutine for each of the three ranges from RARRAY. Subroutine FLOOP performs the summation of Eq. 4 at any range, R, by iteratively calling the FFACTR subroutine for each target height of Table 3. The returned value of F, in decibels, from FFACTR is converted to a pure number, raised to the fourth power, multiplied by the appropriate weighting factor and summed over the height range of the superstructure.

The threshold loop of RTLOOP is used to determine the actual detection range for each of the three thresholds, starting with Lmin. This is done by comparing the one-way propagation loss from Eq. 8 at a range, R, to the applicable threshold. R is decreased, in steps, from the maximum range of 370 km (200 nmi) to r_d to r_o, and from r_o in 1-km steps. At each step, the propagation loss from Eq. 8 is compared to the threshold value. If the threshold at any range R is greater than, or equal to, the one-way loss, then the actual detection range is greater than, or equal to, R. Once the threshold exceeds the one-way loss, the exact detection range is determined by iteratively halving the difference between the two range values that bound the threshold until the range difference between iteration steps is less than 0.1 km. When this occurs, the last range used is declared the detection range for that threshold, and this value is stored in an array to be printed when all five targets have been processed. This procedure is then repeated for the other two thresholds and repeated again for the other ship targets. A minimum range of 5.0 km (2.7 nmi) is the shortest range allowed as a valid detection range for any threshold or target. Additionally, all calculated detection ranges are limited to the maximum instrumented range of the radar.

Figure 3 illustrates several one-way propagation-loss plots from Eq. 8 for a C-band (5600 MHz) radar and the cruiser-class target of Table 3. Figure 3 plots propagation loss versus range for the three different environmental conditions: a standard atmosphere, a 10-meter evaporation duct, and a 300-meter surface-based duct. The radar antenna height is 30.5 meters, the free-space range for a 1 square-meter target is 17.3 nmi, and the maximum instrumented range is 125 nmi. The detection range for each of the various thresholds occurs at the range where the propagation-loss curve crosses the horizontal threshold line. A propagation-loss plot for a point-source target (dotted line) located at 15.5 meters and a standard atmosphere is also included for comparison. Within the optical interference region, the point-source target has deep multipath nulls which the distributed target does not have, a consequence of averaging over the superstructure of the ship. r_o and r_d for the 15.5-meter target height are plotted for reference. The effect of the environment on detection range is quite evident. The detection range is dramatically increased for both the evaporation duct and the 300-meter surface-based duct. In the latter case, the target should be detected to the maximum instrumented range of the radar for all detection thresholds.

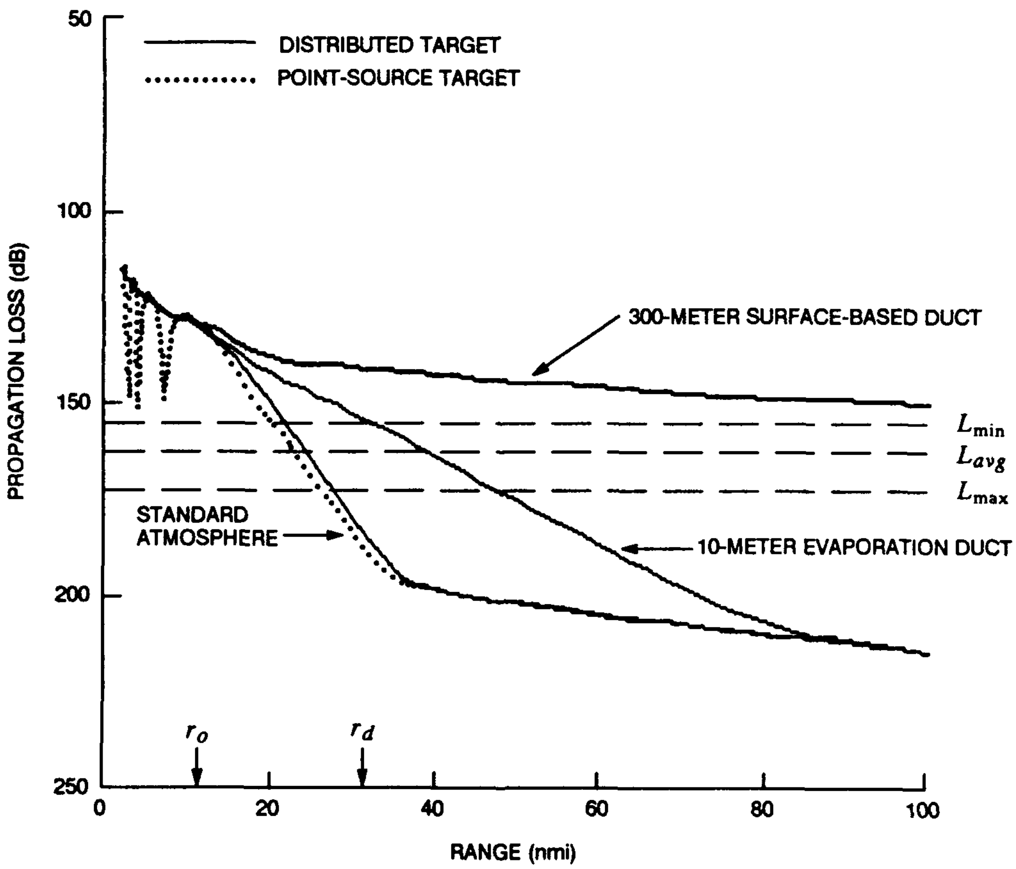

Figure 3. One-way propagation loss for a C-band radar for a cruiser-size target.

4.0 TEST CASES

A number of radar system and environmental inputs are required to determine detection ranges for surface search radar systems. Table 4 lists radar system parameters for two test case systems, Sys 1 and Sys 2, which are used to verify the proper operation of the SSRDRT program. Three different environmental test case conditions, Env1 through Env3, are listed in Table 5. The variable names of sections 2.0 and 3.0 are used in the tables. The environment of Env1 corresponds to a standard atmosphere *M*-unit profile, a 0-meter evaporation duct height, and 10 knots of wind. Env2 uses the same *M*-unit profile as Env1, but the wind speed is 0 knots and a 10-meter evaporation duct is present. Env3 has an *M*-unit profile that contains a 100-meter surface-based duct, a 0-meter evaporation duct, and 5 knots of wind. Both of the radar system test cases use each of the environments.

Table 4. Radar system test set input data.

Parameter	Sys1	Sys2
f, MHz	5600.0	9525.0
H_t, m	30.5	24.4
R_1, nmi	17.3	12.3
R_i, nmi	125.0	125.0
Polarization	Horizontal	Horizontal
Antenna Type	Sin$(x)/x$	Sin$(x)/x$
BW, deg	12.0	18.0
μ_o, deg	0.0	0.0

Table 5. Environmental test set input data.

Parameter	Env1	Env2	Env3
δ, m	0.0	10.0	0.0
W_s, kt	10.0	0.0	5.0
H_1, M_1, (m, M)	(0.0, 339.0)	(0.0, 339.0)	(0.0, 350.0)
H_2, M_2, (m, M)	(1000.0, 457.0)	(1000.0, 457.0)	(270.0, 381.9)
H_3, M_3, (m, M)	(10,000.0,1519.0)	(10,000.0,1519.0)	(300.0, 340.0)
H_4, M_4, (m, M)	n/a	n/a	(1000.0, 422.6)
H_5, M_5, (m, M)	n/a	n/a	(10,000.0,1484.6)

Tables 6 through 11 list the expected output data for the different radar/environmental test cases. The outputs, in nautical miles, are listed to the nearest 0.1 nmi, and the SSRDRT program is considered to be operating correctly if the output is within 0.1 nmi of the value listed in the appropriate table.

Table 6. Output data for Sys 1 and environment 1.

Target	Detection Range (nmi)		
	Minimum	Average	Maximum
Carrier	25.3	28.0	32.4
Cruiser	21.7	24.3	28.7
Destroyer	20.6	23.2	27.5
Frigate	18.8	21.4	25.7
Patrol Boat	12.9	15.3	19.5

Table 7. Output data for Sys 1 and environment 2.

Target	Detection Range (nmi)		
	Minimum	Average	Maximum
Carrier	39.8	46.6	58.0
Cruiser	32.0	38.9	50.0
Destroyer	29.5	36.3	47.5
Frigate	27.2	34.2	45.2
Patrol Boat	15.2	22.0	33.4

Table 8. Output data for Sys 1 and environment 3.

Target	Detection Range (nmi)		
	Minimum	Average	Maximum
Carrier	125.0	125.0	125.0
Cruiser	125.0	125.0	125.0
Destroyer	122.9	125.0	125.0
Frigate	97.1	125.0	125.0
Patrol Boat	21.8	68.6	125.0

Table 9. Output data for Sys 2 and environment 1.

Target	Detection Range (nmi)		
	Minimum	Average	Maximum
Carrier	24.0	26.3	30.1
Cruiser	20.7	23.0	26.9
Destroyer	19.5	21.9	25.7
Frigate	17.9	20.2	24.1
Patrol Boat	12.2	14.4	18.2

Table 10. Output data for Sys 2 and environment 2.

Target	Detection Range (nmi)		
	Minimum	Average	Maximum
Carrier	60.1	72.7	93.6
Cruiser	48.6	60.9	81.5
Destroyer	44.8	57.0	77.5
Frigate	42.4	54.6	75.0
Patrol Boat	26.5	37.9	57.6

Table 11. Output data for Sys 2 and environment 3.

Target	Detection Range (nmi)		
	Minimum	Average	Maximum
Carrier	125.0	125.0	125.0
Cruiser	125.0	125.0	125.0
Destroyer	93.3	125.0	125.0
Frigate	73.8	125.0	125.0
Patrol Boat	18.4	52.1	125.0

5.0 REFERENCES

Anderson, K.D., "Surface-Search Radar Performance in the Evaporation Duct: Global Predictions," Naval Ocean Systems Center Technical Report 923, Revision A, San Diego, CA, May 1986.

Hattan, C.P., "Specification for Radar Free-Space Detection Range and Free-Space Intercept Range Calculations," Naval Ocean Systems Center Technical Document 1598, San Diego, CA, July 1989.

Hattan, C.P., "Specification for a Standard Electromagnetic Propagation Model," Naval Ocean Systems Center Technical Document 1772, San Diego, CA, June 1990.

Jane's Fighting Ships 1980-81, John Moore, ed., S. Low, Marston and Co., Ltd., London, 1980.

Kerr, D.E., *Propagation of Short Radio Waves*, McGraw-Hill Book Company, Inc., New York, 1951.

Patterson, W.L., "Effective Use of the Electromagnetic Products of TESS and IREPS," Naval Ocean Systems Technical Document 1369, San Diego, CA, October 1988.

Queen, F.D. and E.E. Maine, "Radar Cross Sections of Surface Ships at Grazing Incidence," Naval Research Laboratory Report 7388, November 18, 1971. (Distribution limited to U.S. Government agencies.)

Skolnik, M.I., "An Empirical Formula for the Radar Cross Section of Ships at Grazing Incidence," IEEE Trans. Aerospace and Electronic Systems, March 1974.

Appendix
SURFACE-SEARCH RADAR-DETECTION-RANGE-MODEL PROGRAM LISTING

```
c
c **************************** MAIN PROGRAM**************************
c
c   MAIN is a driver program that can be used to demonstrate the
c   correct operation of the Surface Search Radar Detection Range
c   Table (SSRDRT) program.  Main allows the operator to input
c   radar system and environmental inputs from files or keyboard.
c   Main also prints the completed range table from SSRDRT. This
c   program is NOT intended as part of the SSRDRT program, nor are
c   the ENVINP, ENVFIL, SYSINP, SYSFIL and PRNRNG programs.
c
c
c INPUTS:       VARIABLE    VARIABLE DESCRIPTION    (VALID RANGE, UNITS)
c
c EM SYSTEM:    freq        SYSTEM FREQUENCY           (100 - 20000 MHz)
c               ht          TRANSMITTER ANTENNA HEIGHT      (1 - 100 m)
c               fsrng       RADAR FREE SPACE RANGE FOR 1 SQ. m TARGET
c                           IN nmi.                        (1 - 1000 nmi)
c               rinst       MAXIMUM INSTRUMENTED RANGE OF RADAR, nmi
c                           IN nmi.                         (10 -200 nmi)
c               polar       ANTENNA POLARIZATION    (HORIZONTAL = "H",
c                                      VERTICAL = "V", CIRCULAR = "C")
c               antype      ANTENNA TYPE       (OMNIDIRECTIONAL = "O",
c                              SIN(X)/X = "S", COSECANT-SQUARED = "C",
c                                             GENERIC HT-FINDER = "H")
c               bwidth      ANTENNA BEAM WIDTH              (.5 - 45 DEG)
c               elevat      ANTENNA ELEVATION ANGLE     (-10 - +10 DEG)
c                             (0 DEGREES IS HORIZONTAL, NORMAL POINTING
c                              ANGLE FOR SHIPBOARD RADAR SYSTEMS)
c
c ENVIRONMENTAL:
c               delta       EVAPORATION DUCT HEIGHT            (0 - 40 m)
c               height(i)   HEIGHT ARRAY IN METERS - UP TO 30 ELEMENTS
c               munits(i)   M-UNIT ARRAY CORRESPONDING TO HEIGHT ARRAY
c               wind        WIND SPEED                     (0 - 50 KNOTS)
c
c PROGRAM OUTPUT:
c               rng(5,3)    ARRAY OF RANGES FOR 5 CLASSES OF SHIP TARGETS
c                           AND THREE RADAR CROSS-SECTIONS FOR EACH OF
c                           THE TARGETS.  THE CROSS SECTION VARIATIONS
c                           REPRESENT DIFFERENT VIEWING ASPECT.
c
c
```

```
      include 'envsys.common'
      include 'ffac.common'
      include 'surf.common'
      real*4 dmdh(32), hmrs(32)
      integer*2 ntot
c
c
c       Enter the environmental and EM system parameters.
c
      call envinp(DELTA, HEIGHT, MUNITS, WIND, NMAX)
      call sysinp(FREQ, HT, FSR1SM, RINST, POLAR, ANTYPE,
     1            BWIDTH, ELEVAT)
c
c       Calculate surface-search range tables.
c
      call ssrdrt
c
c       Print range table values.
c
      call prnrng
c
c
      END

c
c  Subroutine SSRDRT
c
c  SSRDRT is used to calculate a table of detection ranges for a
c  given radar system and environmental propagation condition. A
c  table of detection ranges will be calculated for 5 ship targets:
c  an aircraft carrier, a cruiser, a destroyer, a frigate and a
c  patrol boat.  Three ranges will be calculated for each target,
c  the ranges correspond to the radar-cross section fluctuations
c  as a function of viewing angle (aspect).
c
c  Variable:       Description:
c
c    alphac     Critical angle - 1st angle not trapped in a
c                 surface-based duct.
c    antbwr     Antenna beam width in radians.
c    antelr     Antenna elevation angle in radians.
c    antfac     Antenna pattern constant.
c    antype     Antenna type:   O - omnidirectional
c                               C - cosecant-squared
c                               H - generic height-finder
c                               S - sin(x)/x
c    bwidth     Antenna beam width in degrees.
c    dffac      Diffraction field constant in dB.
c    dMdh       Array containing M-unit gradients.
c    elevat     Antenna elevation angle in degrees.
```

```
c     elmaxr      Maximum elevation angle in radians.
c     freq        Radar system frequency in MHz.
c     fzt         Height-gain for the transmitter height, in dB.
c     ht          Transmitter antenna height in m.
c     height      Array containing environmental input height values
c                   corresponding to the Munits array.
c     hmrs        Height array containing the elements of the height
c                   array and ht.
c     Munits      Array containing the environmental input M-unit values.
c     nmax        Maximum number of layers in the height & Munit arrays.
c     ntot        Maximum number of layers in the hmrs & dMdh arrays.
c     patrfac     Antenna pattern constant.
c     rk          Effective earth radius factor.
c     sbdht       Height of the surface-based duct in meters.
c
c
      subroutine ssrdrt
c
      include 'envsys.common'
      include 'ffac.common'
      include 'surf.common'
      real*4 dmdh(32), hmrs(32)
      integer*2 ntot
c
c       Test input profile for surface-based ducts, calculate
c       critical angle and provide dMdh, hmrs arrays.
c
      call mprof(height, Munits, ht, NMAX, ALPHAC, DMDH, HMRS,
     1           SBDHT, NTOT)
c
c       Calculate the effective earth radius factor
c
      call getk(alphac, dMdh, hmrs, ntot, ht, RK)
c
c       Initialize antenna parameters.
c
      call antpar(antype, bwidth, elevat, ANTBWR, ANTELR, ANTFAC,
     1             ELMAXR, PATRFAC)
c
c       Calculate diffraction and optical region constants.
c
      call opcnst
c
      call dconst
      IF (sbdht .EQ. 0.0) THEN
        call hgain(ht, FZT)
        dffac = dffac - fzt
      END IF
c
```

```
c        Calculate surface-search range tables.
c
      call rtloop
c
c
      END

c
c  TARGET block data statements
c
c  Target contains the data for the target height (hgt) and weight
c  (wgt) arrays.  Also the target vessel class displacement in
c  kilotons to the 3/2 power.
c
c  VARIABLE:        DESCRIPTION:
c    disp         Target vessel displacement in kilotons, raised to
c                 the 3/2 power.
c    hgt(i,j)     Array containing the height intervals to be used in
c                 a piecewise integration of the received power from
c                 each of the selected target vessels, m.
c    ielem        Maximum number of elements for each of the selected
c                 target classes.
c    wgt(i,j)     Weighting factor array, each element corresponding to
c                 a like-numbered height array element.  The sum of the
c                 i-elements of wgt(*,i) is equal to 1.0, for each
c                 target.
c
c
      BLOCK DATA target
c
c
      include 'surf.common'
c
c
      DATA disp /788.65, 32.15, 11.18, 6.27, 0.0962/
      DATA ielem /26, 31, 31, 22, 10/
c
c
c     Target height and weight array data for Kitty Hawk class CV.
c
      DATA (hgt(1,i), i=1,26)/17.50,  18.50,  19.50,  20.50, 21.50,
     2                        22.50,  23.50,  24.50,  25.50, 26.50,
     3                        27.50,  28.50,  29.50,  30.50, 31.50,
     4                        32.50,  33.50,  34.50,  35.50, 36.50,
     5                        37.50,  38.50,  39.50,  40.50, 41.50,
     6                        42.50/
      DATA (wgt(1,i), i=1,26)/0.0180, 0.0180, 0.0436, 0.0821, 0.0821,
     2                        0.1113, 0.1186, 0.1151, 0.1010, 0.1010,
     3                        0.0645, 0.0402, 0.0402, 0.0094, 0.0094,
     4                        0.0080, 0.0058, 0.0058, 0.0058, 0.0058,
```

```
     5                        0.0052, 0.0028, 0.0028,0.0017, 0.0010,
     6                        0.0010/
c
c     Target height and weight array data for a Virginia class CG.
c
      DATA (hgt(2,i), i=1,31)/ 7.50,   8.50,   9.50,  10.50, 11.50,
     2                        12.50,  13.50,  14.50,  15.50, 16.50,
     3                        17.50,  18.50,  19.50,  20.50, 21.50,
     4                        22.50,  23.50,  24.50,  25.50, 26.50,
     5                        27.50,  28.50,  29.50,  30.50, 31.50,
     6                        32.50,  33.50,  34.50,  35.50, 36.50,
     7                        37.50/

      DATA (wgt(2,i), i=1,31)/0.0151, 0.0151, 0.0151, 0.0634, 0.0688,
     2                        0.0688, 0.0933, 0.0995, 0.0995, 0.0891,
     3                        0.0847, 0.0847, 0.0541, 0.0337, 0.0337,
     4                        0.0208, 0.0079, 0.0079, 0.0067, 0.0049,
     5                        0.0049, 0.0049, 0.0049, 0.0049, 0.0044,
     6                        0.0024, 0.0024, 0.0022, 0.0008, 0.0008,
     7                        0.0008/
c
c     Target data for a Farragut class DD.
c
      DATA (hgt(3,i), i=1,31)/ 5.50,   6.50,   7.50,   8.50,  9.50,
     2                        10.50,  11.50,  12.50,  13.50, 14.50,
     3                        15.50,  16.50,  17.50,  18.50, 19.50,
     4                        20.50,  21.50,  22.50,  23.50, 24.50,
     5                        25.50,  26.50,  27.50,  28.50, 29.50,
     6                        30.50,  31.50,  32.50,  33.50, 34.50,
     7                        35.50/

      DATA (wgt(3,i), i=1,31)/0.0151, 0.0151, 0.0151, 0.0634, 0.0688,
     2                        0.0688, 0.0933, 0.0995, 0.0995, 0.0891,
     3                        0.0847, 0.0847, 0.0541, 0.0337, 0.0337,
     4                        0.0208, 0.0079, 0.0079, 0.0067, 0.0049,
     5                        0.0049, 0.0049, 0.0049, 0.0049, 0.0044,
     6                        0.0024, 0.0024, 0.0022, 0.0008, 0.0008,
     7                        0.0008/
c
c     Target data for an Garcia class FF.
c
      DATA (hgt(4,i), i=1,22)/  5.50,   6.50,   7.50,  8.50,   9.50,
     2                         10.50,  11.50,  12.50, 13.50,  14.50,
     3                         15.50,  16.50,  17.50, 18.50,  19.50,
     4                         20.50,  21.50,  22.50, 23.50,  24.50,
     5                         25.50,  26.50/

      DATA (wgt(4,i), i=1,22)/0.0212, 0.0212, 0.0818, 0.0970, 0.1229,
     2                        0.1402, 0.1318, 0.1193, 0.1049, 0.0475,
     3                        0.0475, 0.0111, 0.0111, 0.0077, 0.0069,
     4                        0.0069, 0.0069, 0.0055, 0.0034, 0.0029,
```

```
     5                        0.0011, 0.0011/
c
c     Target data for an OSA class patrol boat.
c
      DATA (hgt(5,i), i=1,10)/  2.50,    3.50,    4.50,   5.50,    6.50,
     2                          7.50,    8.50,    9.50,  10.50,   11.50/
      DATA (wgt(5,i), i=1,10)/0.0467, 0.2134, 0.3083, 0.2625, 0.1045,
     2                        0.0245, 0.0151, 0.0151, 0.0074, 0.0025/
c
c
      END
c
c     'envsys.common' include file
c
c     EM system parameter common blocks
c
      common  / emsystem  / freq, hr, ht
      common  / emsystem  / polar, antype, bwidth, elevat
c
c     Environmental parameter common blocks
c
      common  / enviro    / delta, height, Munits, nmax, wind
c
c
      real*4 delta, height(30), Munits(30), wind
      real*4 freq, ht, hr, bwidth, elevat
      character*1 antype, polar
      integer*2 nmax
c
c
c     'ffac.common' include file
c
      common  / comffactr / ae, ae2, aeth, alpha, alphac, antbwr
      common  / comffactr / antelr, antfac, atten
      common  / comffactr / c1, c2 ,c3 ,c4 ,c5, c6, c7
      common  / comffactr / del, dffac, difac, elmaxr, exloss
      common  / comffactr / fsterm, hbar, hbfreq, hdif, hmin, h1
      common  / comffactr / h2, horznl, patd, patrfac, rk, rkmin
      common  / comffactr / rnimag, rnreal, rsdfac, rsubd, sbdht
      common  / comffactr / thefac, twoae, zfac, zmax

      real*4 ae, ae2, aeth, alpha, alphac, antbwr, antelr, antfac,
     1       atten, c1, c2, c3, c4, c5, c6, c7, del, dffac, difac,
     2       elmaxr, exloss, fsterm, hbar, hbfreq, hdif, hmin,
     3       horznl, h1, h2, patd, patrfac, rk, rkmin, rnimag,
     4       rnreal, rsdfac, rsubd, sbdht, thefac, twoae, zfac, zmax
c
c
c     'surf.common' include file
```

```
c
      common  / target /  ielem(5), disp(5), hgt(5,31), wgt(5,31)
      common  / radar /   fsrlsm, rinst
      common  / range /   rng(3,5)

      real*4  disp, fsrlsm, hgt, rng, rinst, wgt
      integer*2 ielem
c

c
c  Subroutine ANTPAR
c
c  ANTPAR is used to intitialize antenna parameters for use in
c  calculating antenna pattern factors.
c
c  Variable:        Description:
c
c    Antbwr      Antenna beam width in radians.
c    Antelr      Antenna elevation angle in radians.
c    Antfac      Antenna pattern constant.
c    Antype      Antenna type:  O - omnidirectional
c                               S - Sin(x)/x
c                               C - Cosecant-squared
c                             H - generic Height-finder
c    Bwidth      Antenna beam width, degrees.
c    Elevat      Antenna elevation angle, degrees.
c    Elmaxr      Maximum angle in main beam of antenna, radians.
c    Patrfac     Pattern factor constant for Sin(x)/x antennas,
c                  used to calculate Elmaxr for Sin(x)/x antennas.
c
      SUBROUTINE antpar(antype,bwidth,elevat,ANTBWR,ANTELR,ANTFAC,
     1                   ELMAXR,PATRFAC)
c
c
      real*4 antbwr, antelr, antfac, amax, bwidth, elmaxr,
     1       elevat, pi, patfac, patrfac
      character*1 antype
c
      PI = 3.14159
c      Convert beam width and elevation angle to radians.
      antbwr = 1.745e-2*bwidth
      antelr = 1.745e-2*elevat
      elmaxr = 1.047
      IF (antype .NE. "O") THEN
        IF (antype .EQ. "C") THEN
c           Cosecant-squared antenna pattern constants.
          elmaxr = antelr + .78525
          antfac = SIN(antbwr)
        ELSE
          IF ((antype .EQ. "S").OR.(antype .EQ. "H"))THEN
```

```
c           Sin(x)/x and height-finder antenna pattern constants.
          antfac = 1.39157/SIN(antbwr/2.0)
          amax = PI/antfac
          patrfac = -ATAN(amax/SQRT(1.0 - amax*amax))
          IF (antype .EQ. "S") elmaxr = antelr - patrfac
        END IF
      END IF
    END IF
    RETURN
    END

c
c  Subroutine ANTPAT
c
c  ANTPAT returns the antenna pattern factor for a given angle
c  and antenna type.
c
c  Variable:      Description:
c    alpha      Direct ray launch angle, radians.
c    antbwr     Antenna beam width in radians.
c    antelr     Antenna elevation angle in radians.
c    antfac     Pattern constant.
c    angle      The angle for which the pattern factor is desired.
c    antype     Antenna pattern type: O - omnidirectional
c                                     S - sin(x)/x
c                                     C - cosecant-squared
c                                     H - generic height-finder
c    patfac     The antenna pattern factor for the given angle.
c    patrfac    Pattern constant.
c
c
      SUBROUTINE antpat(antype,alpha,antbwr,antelr,antfac,patrfac,
     1                  angle,PATFAC)
c
c
      real*4 alpha, alpha0, antbwr, antelr, antfac, angle, apat,
     1       patfac, patrfac, ufac
      character*1 antype
c
       patfac = 1.0
       IF (antype .NE. "O") THEN
c           Antenna types other than omni require calculation.
         IF ((antype .EQ. "H").AND.(alpha .GT. antelr)) THEN
           alpha0 = alpha
         ELSE
           alpha0 = antelr
         END IF
         apat = angle - alpha0
         IF (antype .EQ. "C") THEN
c               Cosecant-squared antenna type.
```

```
          patfac = AMIN1(1.0, AMAX1(0.03, 1.0 + apat/antbwr))
          IF (apat.GT.antbwr) patfac = SIN(antbwr)/SIN(ABS(apat))
        ELSE
c               SIN(X)/X antenna type.
          IF (apat .NE. 0.0) THEN
             IF ((angle .LE. alpha0+patrfac).OR.
     1          (angle .GE. alpha0-patrfac)) THEN
c               Antenna pattern is limited to main lobe only.
               patfac = 0.03
             ELSE
c               Sin(x)/x calculation.
               ufac = antfac*SIN(apat)
               patfac = AMIN1(1.0, AMAX1(0.03, SIN(ufac)/ufac))
             END IF
          END IF
        END IF
      END IF
      RETURN
      END

c
c  Subroutine DCONST
c
c  DCONST initializes variables for the diffraction and troposcatter
c  region routines.
c
c  Variable:     Description:
c    arg        Evaporation duct model temporary variable.
c    atten      Diffraction region attenuation rate in dB/km.
c    c1 - c7    Evaporation duct constants for height-gain function.
c    del        Scaled evaporation duct height (delta * zfac).
c    delta      Evaporation duct height, m.
c    dffac      Diffraction field constant in dB.
c    fmax       Evaporation duct model temporary variable.
c    freq       EM system frequency in MHz.
c    fsterm     Free-space loss term, dB.
c    gamma      Evaporation duct excitation factor in dB.
c    hmin       Minimum allowable height, m.
c    rfac       Evaporation duct range scale factor.
c    rk         Effective earth radius factor.
c    rkmin      Minimum rk used for calculation of diffraction field
c               minimum range, rsubd.
c    rsdfac     Constant used for calculation of rsubd, km.
c    sbdht      Surface-based duct height, m.
c    zfac       Evaporation duct height scale factor.
c    zmax       Evaporation duct height variable.  Height where the
c                 two different equations for the height-gain factors
c                 must be equal (del >= 10.25 meters).
c
      SUBROUTINE dconst
```

```
c
c
      real*4 arg, fmax, gamma, rfac, slope
c
      include 'ffac.common'
      include 'envsys.common'
c
c
      IF (sbdht .GT. 0.0) THEN
c
c         Surface-based duct model.
c
        del = 0.0
        hmin = 1.0
        atten = 0.0
        dffac = fsterm
c
      ELSE
c
c         The following terms are for NOSC evap duct model.
c
        rfac = 0.04705 * freq**(1./3.)
        zfac = 0.002214 * freq**(2./3.)
        hmin = 1.0
        del = AMIN1(delta * zfac, 23.3)
        IF (del .GE. 10.25) THEN
c           Constants for scaled evap. duct heights >= 10.25 meters.
          c1 = -0.1189 * del + 5.5495
          c3 = 3./2.
          c2 = 1.3291 * SIN(0.218 * (del-10.0)**0.77) + 0.2171*ALOG(del)
          c2 = c2 * 4.72**(-c3)
          c4 = 87.0 - SQRT(313.29 - (del - 25.3)**2)
          zmax = 4.0 * EXP(-0.31*(del - 10.0)) + 6.0
          arg = c2 * zmax**c3
          slope = 4.72 * c1 * c2 * c3 * SQRT(zmax) / TAN(arg)
          c7 = 49.4 * EXP(-0.1699*(del - 10.0)) + 30.0
          fmax = c1 * ALOG(SIN(arg)) + c4 - c7
          c6 = (zmax/4.72) * slope / fmax
          c5 = fmax / zmax**c6
        ELSE
c           Constants for scaled evap. duct heights <= 10.25 meters.
          c2 = SQRT(40623.61 - (del + 4.4961)**2) - 201.0128
          c1 = (-2.2 * EXP(-0.244*del) + 17.0)*4.72**(-c2)
          c4 = SQRT(14301.2 - (del + 5.32545)**2) - 119.569
          c3 = (-33.9 * EXP(-0.5170001*del) - 3.0)*4.72**(-c4)
          c5 = 41.0 * EXP(-0.41*del) + 61.0
        END IF
        atten = 92.516 - SQRT(8608.7593 - (del - 20.2663)**2)
        IF (atten .LT. 0.0009) atten = 0.0009
```

```
        atten = atten * rfac
        IF (del .LE. 3.8) gamma = 216.7 + del * 1.5526
        IF (del .GT. 3.8) gamma = 222.6 - (del - 3.8) * 1.1771
        dffac = 51.1 + gamma + 10.0 * ALOG10(rfac)
      END IF
c
c        Constants used to calculate rsubd, the range at which
c        the diffraction field solutions are valid.
c
      rkmin = AMAX1(rk, 1.3333)
      rsdfac =  230.2 * (rkmin**2 / freq)**(1.0/3.0)
c
      RETURN
      END

c
c  Subroutine DIFF
c
c  Subroutine DIFF returns the diffraction field propagation factor
c  as a function of range.
c
c  VARIABLES:       DESCRIPTION:
c    atten        NOSC model attenuation rate in dB/km
c    delta        Evaporation duct height in meters
c    dfloss       - 20*LOG(F), where F is the propagation factor
c    dloss        Diffraction field strength in dB
c    dif          Temporary variable
c    difac        NOSC evaporation duct model constant
c    diffe        NOSC evaportion duct model loss in dB
c    exloss       Antenna loss for lowest angle in optical region (dB)
c    r            Range in km
c    tloss        Troposcatter loss from Tropo Subroutine in dB
c
c
      SUBROUTINE diff(r, DFLOSS)
c
c
      real*4  dif, dfloss, dloss, diffe, r, tloss, tlr
c
      include 'ffac.common'
      include 'envsys.common'
c
      tlr = 10.0*ALOG10(r)
      IF (sbdht .EQ. 0.0) THEN
c             Calculate the evaporation duct loss.
        dloss = difac + tlr + atten*r
      ELSE
c             Calculate the surface based duct loss.
        dloss = difac + 2.0*tlr
      END IF
```

```
      dloss = dloss + exloss
c
c       Calculate troposcatter loss and compare to dloss. If the
c       difference is +/- 18 dB add the two fields togather.
c
      call tropo(r,tloss)
      dif = dloss - tloss
      IF (dif .GE. 18.0) THEN
c        Troposcatter field dominates.
         dloss = tloss
      ELSEIF (dif .GE. -18.0) THEN
c        Add troposcatter and diffractions fields togather.
         dloss = dloss - 10.0*ALOG10(1.0 + 10.0**(dif/10.0))
      END IF
c
c      -20*LOG(F) = actual loss - free space loss
c
      dfloss = dloss - fsterm - 2.0*tlr
      RETURN
      END

c
c  Subroutine DUCTS
c
c  DUCTS builds an array containing the top, bottom, and
c  minimum refractivity of all the major ducts in the
c  atmosphere refractivity profile.
c
c  Variable:        Description:
c    dct          3,*  duct parameters array.
c                   1,n  bottom of duct 'n', meters.
c                   2,n  top of duct 'n', meters.
c                   3,n  minimum refractivity of duct 'n', M-units.
c    lvls         Number of refractivity level in rmu, rhts.
c    ndcts        in:  the maximum number of ducts allowed.
c                 out: the number of ducts found.
c    nq           Duct counter.
c    rht          Height array, meters.
c    rmu          Modified refractivity, M-unit array, elements
c                   correspond to like-number elements of rht array.
c
c
      SUBROUTINE ducts(rmu,rht,lvls,DCT,NDCTS)
c
      real*4 dct,delu,delh,deltu,hbot,htop,rht(32),rmu(32)
      integer*2 lvls,ibot,iduct,iend,iq,itop,ndcts,nq
      dimension dct(3,8)
c
c       Locate all major ducts
      nq=0
```

```
      iq=3*ndcts
      itop=lvls
      iend=ndcts
      ndcts=0
      DO iduct=1,iend
c
c           Look for top of next duct
 1010    continue
         htop=rht(itop)
         if(itop.eq.1) go to 1060
         ibot=itop-1
         if(rmu(itop).le.rmu(ibot)) go to 1020
         itop=itop-1
         go to 1010
c
c           Look fo~ bottom of the duct
 1020    continue
         hbot=rht(ibot)
         if(rmu(ibot).lt.rmu(itop)) go to 1030
         if(ibot.eq.1) go to 1040
         ibot=ibot-1
         go to 1020
c
c           Calculate bottom of duct using linear interpolation
 1030    continue
         delu=rmu(ibot+1)-rmu(ibot)
         delh=rht(ibot+1)-rht(ibot)
         deltu=rmu(itop)-rmu(ibot)
         if(delu.lt.0.01) go to 1040
         hbot=rht(ibot) + deltu*delh/delu
c
c           Store duct parameters in array dct
 1040    continue
         amu=rmu(itop)
         call push(dct,iq,nq,amu)
         call push(dct,iq,nq,htop)
         call push(dct,iq,nq,hbot)
         ndcts=iduct
         itop=ibot
      END DO
c
 1060 continue
      RETURN
      END

c
c  Subroutine ENVFIL
c
c  ENVFIL lists the available environmental files and allows the
c  user to select one.  The selected environmental file is read
```

```
c  and closed.  The data from the file is returned to the calling
c  routine.
c
c  Variable:        Description:
c    delta         Evaporation duct height in m.
c    height        Array of up to 30 elements containing the heights
c                    of the M-unit profile.
c    levels        The number of levels in the height, Munits arrays.
c    Munits        Array of up to 30 elements containing the M-unit
c                    values of the upper-air profile.
c    wind          Wind speed in knots.
c
c
      SUBROUTINE envfil(delta, height, Munits, wind, levels)
c
c
      real*4 delta, height(30), Munits(30), wind
      integer*2 levels, ZR, ZW
      character*12 filename
c
c     Initialize read, write channels
      ZR = 5
      ZW = 6
c
      write (ZW,'(" Available Environmental Files: ")')
c       List all files beginning with "E".
      call system ('ls [E]* 1>&2'//char(0))
      write (ZW,'(//,"Enter input file name: ",$)')
      read (ZR,'(a12)') filename
      open (10,FILE=filename)
c
c       Read wind speed in knots and evaporation duct height in m.
      read (10, '(f4.1)') delta
      read (10, '(f4.1)') wind
c       Read the number of levels in M-unit profile.
      read (10, '(i2)') levels
c       Read the height and M-unit profile array values.
      DO i=1, levels
         read (10, '(2f10.1)') height(i), Munits(i)
      END DO
c       Close environmental file.
      close(10)
c
      RETURN
      END

c
c  Subroutine ENVINP
c
c  Subroutine ENVINP prompts the user to enter environmental parameters
```

```
c  and returns.  Environments can be entered over the keyboard or from
c  a file.  If the environment is entered over the keyboard it can be
c  saved in a file for future use.
c
c  Variable:         Description:
c    delta         Evaporation duct height in m.
c    height        Array of up to 30 elements containing the heights
c                    of the M-unit profile.
c    levels        The number of levels in the height, Munits arrays.
c    Munits        Array of up to 30 elements containing the M-unit
c                    values of the upper-air profile.
c    wind          Wind speed in knots.
c
c
      SUBROUTINE envinp(DELTA, HEIGHT, MUNITS, WIND, LEVELS)
c
      real*4 delta, height(30), Munits(30), wind
      character*20 A, dummy, filename
      integer*2 k, kt, levels, ZW, ZR
c
c       Specify the read (5) and write (6) channel numbers.
      ZW = 6
      ZR = 5
c       Initialize environmental parameters.
      wind = 0.0
      delta = 0.0
      levels = 2
      DO i = 1,30
        height(i) = 0.0
        Munits(i) = 0.0
      END DO
c       Enter the environmental data parameters.
      write(ZW,'("Enter environmental data parameters. You may enter")')
      write(ZW,'("up to 30 layers or enter data from a file.  ")')
c
c       Select environmental file.
      write(ZW,'("Enter data from a file? (yes or no) ",$)')
      read(zr,'(A)')dummy
      IF ((dummy(1:1) .eq. 'y').or.(dummy(1:1) .eq. 'Y')) THEN
        call envfil(delta, height, Munits, wind, levels)
      ELSE
        write(ZW,'("Adjcent layers must have different M-values and")')
        write(ZW,'("at least two layers are required.")')
c
        height(1) = 0.0
        Munits(1) = 0.0
        write(ZW,1000)
 1000   format(/,'Enter M-unit Profile - (Height in meters, M-units)'
     1        /,'Starting height is at surface (0 meters) ')
```

```
c
c   DO loop to enter profile data (Height and Munit arrays).
c
      DO i = 1, 30
 100    write(zw,'(" Enter height in meters (or end) ",$)')
        read(zr,'(A)')dummy
        IF ((dummy(1:1) .EQ. 'e').OR.(dummy(1:1) .EQ. 'E')) goto 200
        k = 1
        kt = 1
        DO WHILE((kt .eq. 1) .and. (k .le. 20))
          IF (dummy(k:k).EQ.' ') dummy(k:k) = '.'
          IF (dummy(k:k).EQ.'.') kt = 0
          k = k + 1
        END DO
        IF (i .gt. 1) THEN
          read(dummy,'(f10.2)')height(i)
          IF (height(i) .LE. height(i-1)) THEN
            write(zw,1010)
1010        format('Heights must increase, re-enter height ')
            goto 100
          END IF
        END IF
        levels = i
        write(zw,'(" Enter M-unit value at level ",$)')
        read(zr,'(A)')dummy
        k = 1
        kt = 1
        DO WHILE((kt .EQ. 1) .AND. (k .LE. 20))
          IF (dummy(k:k).EQ.' ') dummy(k:k) = '.'
          IF (dummy(k:k).EQ.'.') kt = 0
          k = k + 1
        END DO
        read(dummy,'(f10.2)')Munits(i)
        IF ((i .NE. 1) .AND. (Munits(i) .EQ. Munits(i-1))) THEN
          Munits(i) = Munits(i) + 0.1
        END IF
      END DO
 200  continue
      write(ZW,1020)
c
1020  format('Enter evaporation duct height in meters (0 to 40) ',$)
      read(ZR,*) delta
      IF (delta .LT. 0.0) delta = 0.0
      IF (delta .GT. 40.0) delta = 40.0
c
      write(ZW,1030)
1030  format('Enter wind speed in knots (0 to 50) ',$)
      read(ZR,*) wind
      IF (wind .LT. 0.0) wind = 0.0
```

```
      IF (wind .GT. 50.0) wind = 50.0
c
      write(ZW,'("Do you wish to store this environment in a file?",
     1   " (yes or no) ",$)')
      read(zr,'(A)')dummy
      IF ((dummy(1:1) .eq. 'y').or.(dummy(1:1) .eq. 'Y')) THEN
        write (ZW,'(" Current Environmental Files: ")')
        call system ('ls [E]* 1>&2'//char(0))
        write (ZW, 1040)
 1040   format("Enter file name (First letter MUST be E) ",$)
        read (ZR,'(a12)') filename
        open (10,FILE=filename)
c
c     Write wind speed in knots and evaporation duct height in m.
        write(10, '(f4.1)') delta
        write(10, '(f4.1)') wind
c     Write the numbers of levels in M-unit profile.
        write(10, '(i2)') levels
        DO i=1, levels
           write(10, '(2f10.1)') height(i), Munits(i)
        END DO
c         close file
        close(10)
      END IF
c
     END IF
c
     RETURN
     END

c
c  Subroutine FFACTR
c
c  FFACTR returns the value of the pattern propagation factor, F, in dB
c  for specified range , EM system parameters and environmental para-
c  meters.
c
c  Variables:      Description:
c
c    alphac      Critical angle - 1st angle not trapped in surface-
c                  based duct.
c    antbwr      Antenna vertical beamwidth in radians.
c    antelr      Antenna elevation angle in radians.
c    antfac      Antenna pattern constant.
c    antype      Antenna type: O = omnidirectional, S = sin(x)/x,
c                  C = cosecant-squared, H = height-finder.
c    bwidth      Antenna vertical beam width in degrees.
c    delta       Evaporation duct height in meters.
c    deltaf      Variable used in linear interpolation of F in the
c                  intermediate region.
```

```
c    dffac       Diffraction region constant, dB.
c    difac       Diffraction region constant, dB.
c    elevat      Antenna elevation angle in degrees.
c    elmaxr      Maximum elevation angle in main beam of antenna, rad.
c    ff          Pattern propagation factor, F, in dB.
c    freq        EM system operating frequency in MHz.
c    frsubd      Pattern propagation factor at rsubd.
c    fzr         Evaporation duct height-gain function for hr, dB.
c    fzt         Evaporation duct height-gain function for ht, dB.
c    height      Array containing environmental input height values
c                  corresponding to the Munit array.
c    hdif        Height difference between receiver/target and
c                transmitter height in km.
c    ht          Transmitter height in m.
c    hr          Receiver/target height in m.
c    h1          Lower height of hr, ht, in m
c    h2          Higher height of hr, hr, in m.
c    lvlant      Transmitter height level in hmrs and dMdh arrays.
c    Munits      Array containing environmental input M-unit values.
c    nmax        Integer number of layers in Munits and height arrays.
c    opmaxd      Maximum range in the optical interference region, km.
c    opmaxf      F at opmaxd.
c    patrfac     Antenna pattern constant.
c    pd          Path-difference between direct and sea-reflected rays.
c    polar       Antenna polarization: H = horizontal, V = vertical,
c                  C = circular.
c    psi         Grazing angle in radians.
c    r           Range in km.
c    rsdfac      Constant used to calculate rsubd.
c    rsubd       Minimum range where diffraction field solutions are
c                  applicable, km.
c    sbdht       Surface-based duct height, m.
c    theta       Total phase difference between direct and sea-
c                  reflected rays including phase lag due to reflection.
c    wind        Wind speed in kts.

      SUBROUTINE ffactr(r, FF)
c
c
      real*4  deltaf, ff, fzr, fzt, opmaxd, opmaxf, frsubd, r
      real*4  dMdh(32), hmrs(32)
      integer*2 lvlant, ntot
c
      include 'ffac.common'
      include 'envsys.common'
c
c       Call mprof to insert a profile level at Ht and determine if
c       any surface-based ducts are present.  If a surface-based duct
c       is present calculate critical angle, alphac.
```

```
c
c       NOTE: A '*' in column one indicates that line of code has been
c             moved to the main routine to avoid multiple initialization
c             of constants, since FFACTR is called from inside a loop.
c
*     call mprof(height, Munits, ht, NMAX, ALPHAC, DMDH, HMRS,
*    1           SBDHT, NTOT)
c
c       Call getk to determine the effective earth radius factor, rk.
c
*     call getk(alphac, dMdh, hmrs, ntot, ht, RK)
c
c       Define h1, h2 for opticf subroutine.  These are swapped for
c       ht>hr because the iteration loop for r1 in opticf works most
c       efficiently when the lowest height is the transmitter height.
c
      IF (ht .GT. hr) THEN
        h1 = hr
        h2 = ht
      ELSE
        h1 = ht
        h2 = hr
      END IF
c       Define optical region constants.
*     call opcnst
      hdif = (hr - ht) * 1.0e-3
c       Initialize antenna parameters.
*     call antpar(antype,bwidth,elevat,ANTBWR,ANTELR,ANTFAC,
*    1                                                 ELMAXR,PATRFAC)
c       Define diffraction/troposcatter region constants.
*     call dconst
      call hgain(hr, FZR)
*     IF (sbdht .EQ. 0.0) THEN
*       call hgain(ht, FZT)
*       dffac = dffac - fzt
*     END IF
      difac = dffac - fzr
      rsubd = 3.572 * (SQRT(rkmin * hr) + SQRT(rkmin * ht)) + rsdfac
c       Determine maximum range and f-factor in optical region.
      call oplimit(OPMAXD,OPMAXF)
      IF (r .GE. rsubd) THEN
c           Calculate loss for range in diffraction/troposcatter region.
        call diff(r, FF)
      ELSE
        IF  (r .GT. opmaxd) THEN
c             Range is in intermediate region - use linear interpolation
c             on log of the f-factor.
          call diff(rsubd,FRSUBD)
          deltaf = (r - opmaxd) * (opmaxf - frsubd) / (opmaxd-rsubd)
```

```
            ff = opmaxf + deltaf
          ELSE
c              Range is in the optical interference region.
            IF (r .LE. opmaxd) call opticf(polar,r,PD,PSI,THETA,FF)
          END IF
        END IF
        ff = - ff
        RETURN
        END

c
c  Subroutine FLOOP
c
c  Floop performs the summation of the pattern propagation factors
c  over the height of the target vessel from the main deck to the
c  top of the superstructure.  This is essentially a piecewise in-
c  tegration of the received power from a distributed target at a
c  specified range.
c
c
c  VARIABLE:         DESCRIPTION:
c    ff              Pattern propagation factor, in dB, for range r,
c                    radar system antenna height ht and target height hr.
c    fi4             Pattern propagation factor to the fourth power, f**4,
c                    for the ith height array element.
c    hgt(kn,i)       Array of height elements corresponding to (usually)
c                    1-meter increments of the height of the target
c                    vessel from the main deck to the top of the super-
c                    structure, m.
c    hr              Target height, m.
c    kn              Integer corresponding to a specific ship target.
c                    1 = CV; 2 = CG; 3 = DD; 4 = FF; 5 = Patrol Boat.
c    maxht           Maximum number of elements in the hgt and wgt arrays.
c    r               The range where the summation of f-factors is to be
c                    performed, km.
c    wfi4            Weighted sum of the pattern propagation factors over
c                    the elements of the height array.
c    wgt(kn,i)       Array of weighting factors, each factor corresponding
c                    to the like-numbered hgt(k,i) array element.
c
c
        SUBROUTINE floop(kn, maxht, r, WFI4)
c
        include 'envsys.common'
        include 'ffac.common'
        include 'surf.common'
c
c
        real*4 ff, fi4, r, wfi4
        integer*2 i, kn, maxht
```

```
c
c
      wfi4 = 0.0
      DO i = 1, maxht
c
        hr = hgt(kn,i)
c
c             Sum F-factors
c
        call ffactr(r, FF)
        fi4 = (10.0**(ff/20.0))**4
        wfi4 = wfi4 + wgt(kn,i)*fi4
c
      END DO
c
      RETURN
      END

c
c  Subroutine GETK
c
c  Subroutine GETK is used to determine the effective earth radius
c  factor k.  Getk accomplishes this by tracing a ray from the trans-
c  mitter height to 200 NMi (370 km).  The ray launch angle is 0 deg.
c  if no surface-based duct exists, or alphac, the critical angle if
c  one does.
c
c  Variable:        Description:
c    alphac       Critical angle necessary to escape duct. If alphac
c                   = 0 then no surface-based duct exists.
c    a0           Initial ray launch angle, radians.
c    a1           Ray angle at top of layer, radians.
c    deld         Range difference, km.
c    delh         Height difference, meters.
c    delM         M-unit difference.
c    delmdh       M-unit gradient.
c    dMdh         M-unit gradient array.
c    hlast        Height at 370 km.
c    hmrs         Array of height elements, in meters.
c    ntot         Maximum number of elements in hmrs and dMdh arrays.
c    rdeld        Range incremented in ray trace.
c    rmax         Maximum range for ray trace - 370 km.
c    rng          Range, km.
c    rk           Effective earth radius factor.
c    xmtr         Transmitter height in meters.
c
      SUBROUTINE getk(alphac, dMdh, hmrs, ntot, xmtr, RK)
c
      real*4 alphac, a0, a1, deld, delh, delM, delmdh, dMdh(32)
      real*4 hlast, hmrs(32), rdeld, rmax, rng, rk, xmtr
```

```
      integer*2 ntot, i
c
      rmax = 370.0
      h = xmtr
      rng = 0.0
      a0 = alphac
c        Loop to trace ray through the atmospheric layers.
      DO i=2,ntot-1
        delm = (hmrs(i+1) - h)*dMdh(i)*1.0E-3
        a1 = SQRT(a0*a0 + 2.0*delm)
        deld = (a1 -a0)/dMdh(i)
        rdeld = rng + deld
        IF(rdeld .GT. rmax) GOTO 1000
        a0 = a1
        h = hmrs(i+1)
        rng = rdeld
      END DO
      i = ntot
 1000 continue
c         Ray trace in final layer to range rmax.
      deld = rmax - rng
      a1 = a0 + dMdh(i) * deld
      delM = (a1*a1 - a0*a0)*0.5
      delh = 1000.0*delM/dMdh(i)
      hlast = hmrs(i) + delh
c         Determine the equivalent single-gradient atmosphere that
c         would be required to trace a ray launched at alphac that
c         would arrive at height = hlast at a range of 370 km.
      delmdh = (-alphac)*2.0/rmax + 2.0E-3*(hlast - xmtr)/(rmax*rmax)
      rk = 1.0/(6371.0 * delmdh)
      IF(rk .GT. 5.0) rk = 5.0
      IF(rk .LE. 0.5) rk = 0.50
      RETURN
      END

c
c  Subroutine GTHETA
c
c  GTHETA calculates optical phase-lag difference angle 'theta'
c  between direct and sea-reflected rays using the reflection
c  point range 'r1'
c
c  Variable:       Description:
c    ae2          Effective earth radius * 2000.
c    h1           Height of transmitting antenna, m.
c    h2           Height of receiver/target, m.
c    h1p          Effective height of h1, m.
c    h2p          Effective height of h2, m.
c    plr          Antenna polarization: H = horizontal
c                                       V = vertical
```

```
c                                    C = circular
c    psi         Grazing angle in radians.
c    phi         Phase lag due to reflection, radians.
c    r           Total ground range, km.
c    r1          Reflection point range, (from h1), km.
c    r2          Reflection point range, (from h2), km.
c    rmag        Magnitude of the reflection coefficient.
c    theta       Total phase lag between direct and reflected
c                rays including phi.
c
      SUBROUTINE gtheta(plr,r1,R,THETA,R2,PSI,RMAG)
c
c
      real*4  h1p, h2p, psi, phi, r, r1, r2, rmag, theta
      character*1 plr
c
      include 'ffac.common'
      include 'envsys.common'
c
      h1p = h1 - r1*r1/ae2
      psi = 1.0e-3 * h1p/r1
      IF (psi .GT. 0.3) psi = ATAN(1.0e-3 * h1p/r1)
c       Ray trace equation used to determine r2 based on psi.
      r2 = ( SQRT(psi*psi + 2.0e-3 * h2/ae) - psi ) * ae
      r = r1 + r2
      h2p = h2 - r2*r2/ae2
      call ref(plr,psi,RMAG,PHI)
c       Calculate theta = Path-length difference + phase lag due
c       to reflection (phi).
      theta = phi + thefac*h1p*h2p / r
      RETURN
      END

c  Subroutine HGAIN
c
c  HGAIN returns a height-gain factor in dB for a specified height.
c
c  Variable:       Description:
c    c1 - c7       Constants used to calculate fzdb for evap. ducts.
c    del           Scaled evaporation duct height.
c    delta         Evaporation duct height, m.
c    freq          EM system frequency in MHz.
c    fzdb          Height-gain factor in dB.
c    h             The height for which the height-gain factor is
c                    required, m.
c    hmin          Minimum height.
c    sbdht         Surface-based duct height, m.
c    rfac          Evaporation duct range scale factor.
c    zfac          Evaporation duct height scale factor.
c    zmax          Breakpoint for evporation duct heights > 10.25m.
```

```
c     z1            Scaled height for surface-based ducts.
c     z2            Scaled height for evaporation duct heights.
c
      SUBROUTINE hgain (h, FZDB)
c
      real*4 fzdb, h, z1, z2
c
      include 'ffac.common'
      include 'envsys.common'
c
      fzdb = 0.0
      IF (sbdht .GT. 0.0) THEN
c          Calculate surface-based duct height-gain factor.
        z1 = h / sbdht
        IF ((Freq .LE. 150.0).AND.(z1 .LT. 0.8)) THEN
          fzdb = -60.0 * (z1 - 0.5)**2
        END IF
        IF ((Freq .LE. 150.0).AND.(z1 .GE. 0.8)) THEN
          fzdb = 1.14 * z1**(-6.26) - 10.0
        END IF
        IF ((Freq .GT. 150.0).AND.(z1 .LT. 1.0)) THEN
          fzdb = 10.0 - 200.0 * (z1 - 0.5)**4
        END IF
        IF ((Freq .GT. 150.0).AND.(Freq .LE. 350.0)
     1                               .AND.(z1 .GE. 1.0)) THEN
           fzdb = 7.5 * z1**(-13.3) - 10.0
        END IF
        IF ((Freq .GT. 350.0).AND.(z1 .GE. 1.0)) THEN
          fzdb = 12.5 * z1**(-8.0) - 15.0
        END IF
      ELSE
c         Calculate evaporation duct height-gain factor.
        z2 = AMAX1(h * zfac, hmin)
        IF (Del .GE. 10.25) THEN
c          Calculate height-gain for del>=10.25 meters.
          IF (z2 .GT. zmax) THEN
               fzdb = c5 * (z2**c6) + c7
          ELSE
               fzdb = c1 * ALOG(SIN(c2 * (z2**c3))) + c4
          END IF
        ELSE
c          Calculate height-gain for del<10.25 meters.
          fzdb = (c1 * z2**c2) + (c3 * z2**c4) + c5
        END IF
      END IF
      RETURN
      END
```

```
c
c  Subroutine INSRT
c
c  INSRT inserts (or appends) a new level into the M-unit profile.  It
c  does this by locating the  new height relative to the existing pro-
c  file heights.  If the new height is greater than the top level, then
c  append a new level for the new height.  If the new height is between
c  two levels, then insert a new level for the new height. If the new
c  height is equal to an existing level's height, do not add a new
c  level for the new height.
c
c  Variable:         Description:
c    amu           Modified refractivity array, M-units.
c    hmrs          Height array, meters, each element corresponding to
c                    the like-number amu array element.
c    iq            Number of levels in amu and hmrs.
c    hgt           Height of new level to be added, meters.
c    ipnt          Index pointer to new level.
c
c
      SUBROUTINE insrt(amu,hmrs,iq,hgt,ipnt)
c
      real*4 amu(32),hmrs(32),hgt
      integer*2 iq,ipnt
c
      DO i=1,iq
         ilevel=i
         IF(ABS(hgt-hmrs(ilevel)).LE.0.01) go to 1020
         IF(hmrs(ilevel).GT.hgt) go to 1030
      END DO
c
c        Hgt > amu(iq)
      iq=iq+1
      ipnt=iq
      grdnt=0.1181102
      amu(ipnt)=amu(iq-1) + (hgt-hmrs(iq-1))*grdnt
      hmrs(ipnt)=hgt
      go to 1050
c
c        Hgt = hmrs(ilevel)
 1020 continue
      ipnt=ilevel
c     amu(ipnt)=amu(ilevel)
      hmrs(ipnt)=hgt
      go to 1050
c
c        Hmrs(ilevel) > hgt > hmrs(ilevel-1)
 1030 continue
c        Shift levels above new height up one
```

```
      DO i=ilevel,iq
         j=iq - (i-ilevel)
         hmrs(j+1)=hmrs(j)
         amu(j+1)=amu(j)
      END DO
      iq=iq+1
      ipnt=ilevel
      grdnt=(amu(ipnt+1)-amu(ipnt-1))/(hmrs(ipnt+1)-hmrs(ipnt-1))
      amu(ipnt)=amu(ipnt-1) + (hgt-hmrs(ipnt-1))*grdnt
      hmrs(ipnt)=hgt
c     go to 1050
c
 1050 continue
      RETURN
      END

c
c  Subroutine MPROF
c
c  MPROF modifies the M-unit and height arrays  by inserting a level at
c  the antenna height using straight line interpolation (or a standard
c  atmosphere gradient) to calculate its M-unit value.  The new profile
c  is then used to locate any ducts that might be contained in the pro-
c  file.  If the bottom of the duct is below the EM system antenna
c  height, and the top above the antenna height, then a critical angle
c  is calculated for the EM system in the surface-based duct.  (It is
c  assumed that low-elevated ducts are surface ducts if the EM system is
c  in the duct.)
c
c  Variable:        Description:
c    alphac       The critical penetration angle necessary to escape duct
c    amu          An array of M-unit values
c    antena       EM system antenna height
c    antmu        M-unit value at the EM system antenna height
c    dcts         24  duct parameter array
c                     1,n  bottom of duct 'n', meters
c                     2,n  top of duct 'n', meters
c                     3,n  minimum refractivity of duct 'n', m-units
c    dMdh         M-unit gradient array
c    hbot         Height of the bottom of a duct
c    htop         Height of the top of a duct
c    height       Height array with the original profile heights
c    hmrs         Height array with elements corresponding to the dMdh
c                   array elements
c    lvlant       EM system antenna level
c    lvltop       Maximum number of layers in the hmrs array
c    Munits       M-unit array with elements corresponding to the height
c                   array elements
c    ndcts        The number of ducts stored in 'dcts'
c    nmax         The number of elements in the height and Munit arrays
```

```
c    ntot        The number of elements in the dMdh and hmrs arrays
c    rma         M-unit value at the minimum on the duct profile
c    sbdht       The height of the surface-based duct
c
c    Variables not listed are temporary variables.
c
c
      SUBROUTINE mprof(height,Munits,antena,nmax,ALPHAC,DMDH,HMRS,
     1                   SBDHT,NTOT)
c
c
      real*4 alphac,amu(32),antena,dmdh(32),hmrs(32),height(30)
      real*4 Munits(30),sbdht
      real*4 antmu,dcts,hb,ht,rma
      integer*2 lvlant,lvltop,nmax,ntot
      integer*2 ndcts
      dimension dcts(3,8)
c
      lvltop = nmax
      alphac = 0.0
      sbdht = 0.0
c
c       Copy height and m-unit arrays.
c
      lvltop = nmax
      DO i = 1, nmax
         hmrs(i)=height(i)
         amu(i)=Munits(i)
      END DO
c
c       Insert new level at the antenna height.
c
      call insrt(amu,hmrs,lvltop,antena,lvlant)
      antmu=amu(lvlant)
c
c       Locate all major ducts.
      ndcts=8
      call ducts(amu,hmrs,lvltop,dcts,ndcts)
c
c       Define trapping duct parameters.
      IF(ndcts .NE. 0)THEN
        DO iduct=1,ndcts
          hb=dcts(1,iduct)
          ht=dcts(2,iduct)
          rma=dcts(3,iduct)
          IF((antena .GT. hb) .AND. (antena .LT. ht)) go to 1040
          IF(hb.lt.0.01) go to 1040
        END DO
      END IF
```

```
c
c       Antenna not inside a major duct.
      go to 1050
c
c       The antenna is inside a low-level elevated duct
c          or inside a surface-based duct.
 1040    continue
      sbdht = ht
      alphac=1.0e-3*sqrt(2.0*(antmu-rma)) + 1.0e-5
 1050 continue
c
c     Delete all levels between the surface and the antenna level.
      DO i = lvlant,lvltop
         j=i-(lvlant-2)
         hmrs(j)=hmrs(i)
         amu(j)=amu(i)
      END DO
      lvltop=j
      lvlant=2
c
c       Calculate the M-unit gradient array.
      iend=lvltop-1
      DO i = 1, iend
         delu=amu(i+1)-amu(i)
         delh=hmrs(i+1)-hmrs(i)
         dmdh(i)=1.0e-3*delu/delh
      END DO
      dmdh(lvltop)=0.1181102e-3
c
      ntot = lvltop
      RETURN
      END

c
c  Subroutine OPCNST
c
c  OPCNST initializes optical region constants.
c
c  Variable:       Description:
c    ae           Effective earth radius, (rk * 6371), km.
c    aeth         Effective earth radius * 1000.
c    ae2          Aeth * 2
c    eps          Dielectric constant of sea-water, epsilon.
c    freq         EM system frequency in MHz.
c    fsterm       Free-space loss constant in dB.
c    hbar         RMS wave height due to wind in m.
c    hbfreq       Constant for subroutine ruff,
c                   (hbar * 2 * PI / wavelength).
c    polar        EM system antenna polarization:
c                   H = horizontal, V = vertical, C = circular
```

```
c     rk           Effective earth radius factor.
c     rnreal       Real part of the square of the index of refraction.
c     rnimag       Imaginary part of the square of the index of refract.
c     thefac       Constant used to calculate path-length difference
c                    between direct and sea-reflected rays.
c     twoae        Constant (ae * 2).
c     wind         Wind speed in kts.
c
c
      SUBROUTINE opcnst
c
c
      real*4 eps, sigma
c
      include 'ffac.common'
      include 'envsys.common'
c
      fsterm = 32.44 + 20.0 * ALOG10(freq)
c         Exclusively for REF subroutine
      IF (polar .NE. "H") THEN
c           eps is the permittivity of salt water
c           sigma is the conductivity of salt water
        IF (freq .LE. 1500.0) THEN
          eps = 80.0
          sigma = 4.3
        ELSEIF (freq .LE. 3000.0) THEN
          eps = 80.0 - 0.00733 * (freq - 1500.0)
          sigma = 4.3 + 0.00148 * (freq - 1500.0)
        ELSEIF (freq .LE. 10000.0) THEN
          eps = 69.0 - 0.00243 * (freq - 3000.0)
          sigma = 6.52 + 0.001314 * (freq - 3000.0)
        ELSE
          eps = 51.99
          sigma = 15.718
        END IF
c         Define the real and imaginary parts of the square of
c         the index of refraction of sea-water.
        rnreal = eps
        rnimag = (-18000.0) * sigma/freq
      END IF
c        Define rms wave-height for subroutine RUFF
      hbar = 0.0051 * (0.51477*wind)**2
      hbfreq = 0.02094 * freq * hbar
c
      ae = rk * 6371.0
      twoae = 2.0 * ae
      aeth = rk * 6.371
      ae2  = aeth * 2.0
      thefac = freq * 4.193E-5
```

```
c
      RETURN
      END
c
c  Subroutine OPFFAC
c
c  OPFFAC calculates quantities used to determine the pattern
c  propagation factor (F) in the optical interference region.
c
c  Variable:       Description:
c    ae            Effective earth radius, km.
c    alpha         Direct ray launch angle, radians.
c    angle         Angle for which antenna pattern factor desired.
c    beta          Reflected ray launch angle, radians.
c    divfac        Divergence factor.
c    dr            Constant - product of antenna pattern factor for
c                    reflected ray * divergence factor * reflection
c                    coefficient * surface roughness factor.
c    gamma         Earth's interior angle (r1/ae).
c    psi           Grazing angle, radians.
c    r1            Reflection point range, km.
c    r2            Reflection point range, km.
c    range         Total ground range in km.
c    rmag          Magnitude of reflection coefficient.
c    ruf           Sea-surface roughness coefficient.
c    sinpsi        Sin(psi).
c    twoae         2*ae
c
c
      SUBROUTINE opffac(gamma,range,psi,r1,r2,rmag,ELANG,DPAT,DR)
c
      real*4 angle, beta, divfac, dpat, dr, elang, gamma, psi, r1,
     1       r2, range, rmag, ruf, sinpsi
c
      include 'ffac.common'
      include 'envsys.common'
c
      patfac = 1
c       Calculate direct ray launch angle, alpha.
      alpha = hdif/range - range/twoae
      angle = alpha
      elang = alpha
c       Determine antenna pattern factor for direct ray alpha.
      call antpat(antype,alpha,antbwr,antelr,antfac,
     1                                      patrfac,angle,PATFAC)
      patd = patfac
      dpat = patfac
      beta = - (gamma + psi)
      angle = beta
```

```
c       Determine antenna pattern factor for reflected ray beta.
      call antpat(antype,alpha,antbwr,antelr,antfac,
     1                                       patrfac,angle,PATFAC)
c       Determine surface roughness coefficient.
      sinpsi = SIN(psi)
      call ruff(hbar, hbfreq, psi, sinpsi, RUF)
c       Calculate the divergence factor.
      divfac = 1.0/(SQRT(1.0 + (2.0 * r1 * r2)/(ae * range * sinpsi)))
      dr = patfac * ruf * divfac * rmag
      RETURN
      END

c
c  Subroutine OPLIMIT
c
c  OPLIMIT calculates the maximum range in the optical region, opmaxd,
c  and opmaxl = -20 LOG(F) at opmaxd, where F is the pattern propagation
c  factor.
c
c  Variable:        Description:
c    ae            Effective earth radius, km.
c    ae2           ae * 2000
c    alpha         Direct ray launch angle, radians.
c    alphac        Critical angle in radians.
c    a1            An angle used to determine r1(psilim).
c    exloss        A measure of how much of the antenna's energy
c                    is directed toward the horizon, dB.
c    freq          EM system frequency, MHz.
c    fsqrd         Square of the pattern propagation factor, F.
c    gamma         Earth's interior angle.
c    hdif          Difference in height between h1 and h2.
c    horzn1        Tangent ray distance for height h1, km.
c    h1            Transmitter height, m.
c    h1p           Effective transmitter height, m.
c    h2            Receiver/target height, m.
c    h2p           Effective receiver/target height, m.
c    opmaxd        Maximum range in optical region, km.
c    opmaxl        Propagation factor in dB at opmaxd.
c    pd            Path-length difference between direct and
c                    sea-reflected rays.
c    phi           Phase lag due to reflection from sea-surface.
c    psilim        Grazing angle limit to optical region.
c    psi           Grazing angle in radians.
c    r             Total ground range, km.
c    r1            Reflection point range (from h1), km.
c    r2            Reflection point range (from h2), km.
c    rk            Effective earth radius factor.
c    theta         Total phase difference between direct and sea-reflected
c                    rays (pd) and phase-lag due to reflection, phi.
c    thefac        Constant used to calculate path-lenght difference.
```

```
c     thnext      The next value of theta to be determined.
c
c
      SUBROUTINE oplimit(OPMAXD,OPMAXL)
c
      real*4 al, dr, fsqrd, gamma, halfpi, h1p, h2p, pd, phi,
     1       pi, psi, psilim, r, r1, r2, rmag, theta, thnext
c
      include 'ffac.common'
      include 'envsys.common'
c
      PI = 3.14159
      halfpi = PI / 2.0
      horzn1 = 3.572 * SQRT(rk * h1)
      psilim = 0.01957/(freq*rk)**0.33333
c
c       If both terminals are in the duct set alphac = 0.0
c
      IF ((alphac .GT. 0.0) .AND. (h2 .LT. sbdht)) alphac = 0.0
c
c       Initial guess for r1 is based on grazing angle limit range.
c       Use ray trace equations to determine r1 and r2.
c
      psi = psilim
      a1 = SQRT(psi**2 + 2.0e-3*h1/ae)
      r1 = (a1 - psi)*ae
      r2 = r1
      IF (h2 .GT. h1)r2 = r2 + (SQRT(a1**2 + 2.0*ABS(hdif)/ae) - a1)*ae
      r = r1 +  r2
      h1p = h1 - r1*r1/ae2
      h2p = h2 - r2*r2/ae2
      call ref(polar,psi,RMAG,PHI)
      pd = thefac*h1p*h2p / r
c
c       Calculate theta based on grazing angle limit.
c
      theta = phi + pd
      alpha = hdif/r - r/twoae
      IF (alphac .GT. 0.0) THEN
        IF ((alpha .LT. alphac) .OR. (pd .GT. halfpi)) THEN
c
c           Calculate theta based on range obtained from alphac.
c
          r = (SQRT(alphac**2 + 2.0*ABS(hdif)/ae) - alphac)*ae
          call opticf(polar,r,PD,PSI,THETA,FF)
        END IF
      END IF
      IF ((alphac .GT. 0.0) .AND. (pd .GT. halfpi)) THEN
c
```

```
c       If theta>(2 Pi) then optical limit is 1st peak
c       with theta greater than theta(alphac).
c
        IF (theta .GT. 6.28319) THEN
          thnext = INT(theta/(2.0*PI) + 1)*(2.0 * PI)
          call r1iter(polar,thnext,R1,R2,R,PSI,RMAG)
          theta = thnext
        END IF
      ELSE
c
c       Optical limit is grazing angle limit or 1/4 wavelength limit.
c
        IF ((pd .GT. halfpi) .OR. (psi .NE. psilim)) THEN
c
c       Determine theta value @ 1/4 wavelength limit, (H polar).
c
          thnext = 1.5 * PI
          call r1iter("H",thnext,R1,R2,R,PSI,RMAG)
          IF (polar .NE. "H") THEN
            call ref(polar,psi,RMAG,PHI)
            theta = halfpi + phi
          ELSE
            theta = thnext
          END IF
        END IF
      END IF
      IF (ht .GE. hr) THEN
        gamma = r2/ae
      ELSE
        gamma = r1/ae
      END IF
      call opffac(gamma,r,psi,r1,r2,rmag,ALPHA,PATD,DR)
      fsqrd = (patd*patd + dr*dr + 2.0*dr*patd*COS(theta))
c       Limit fsqrd to prevent runtime errors when taking LOG(fsqrd).
      IF (fsqrd .LT. 1.0e-7) fsqrd = 1.0e-7
      opmaxd = r
      opmaxl = - 10.0 * ALOG10(fsqrd)
      exloss = - 20.0 * ALOG10(patd)
c
      RETURN
      END

c
c  Subroutine OPTICF
c
c  Subroutine OPTICF calculates the total phase difference, theta,
c  between direct and sea-reflected ray paths, including phase
c  change due to reflection from sea-surface.  It then uses theta
c  to determine the value of the pattern propagation factor, F, in
c  the optical region, and returns 20Log(F).
```

```
c
c
c  Variable:        Description:
c    ae            Effective earth radius, km.
c    ae2           Ae*2000, km.
c    aeth          Ae*1000, km.
c    alpha         Direct ray launch angle, radians.
c    dr            Product of divergence factor, surface roughness
c                    coefficient, reflection coefficient and antenna
c                    pattern factor for the reflected ray.
c    epsr          Iteration loop range tolerance, km.
c    ff            Pattern propagation factor, F, in dB.
c    fprl          Value of the derivative of the cubic equation at rl.
c    frl           Value of the cubic equation for a given rl.
c    fsqrd         Square of the pattern propagation factor.
c    gamma         Earth's interior angle (rl/ae) in radians.
c    hrp           Effective receiver/target height, m.
c    htp           Effective transmitter height, m.
c    h1            The transmitter height, m.
c    h2            The receiver/target height, m.
c    pd            The path-length difference between direct and re-
c                    flected rays in radians.
c    phi           Phase lag due to reflection from sea surface, rad.
c    psi           Grazing angle in radians.
c    r             Total ground range, km.
c    rl            Reflection point range, (from xmtr), km.
c    rlsqrd        Square of the reflection point range.
c    r2            Reflection point range, (from rcvr/target), km.
c    rr            Iteration loop variable - range difference.
c    rmag          Magnitude of reflection coefficient.
c    t             Iteration loop variable.
c    theta         Total phase lag between direct and sea-reflected
c                    rays, in radians.  (theta = pd + phi)
c    thefac        Constant used to calculate theta.
c    v             Iteration loop variable.
c    w             Iteration loop variable.
c
c
      SUBROUTINE opticf(plr,r,PD,PSI,THETA,FF)
c
      real*4 dr, epsr, ff, frl, fprl, fsqrd, gamma, hrp, htp,
     1       phi, psi, r, rl, rlsqrd, r2, rmag, rr, t, theta, v, w
      character*1 plr
      integer*2 jk
c
      include 'ffac.common'
      include 'envsys.common'
c
      rl = (h1/(h1 + h2))*r
```

```
      t = -1.5 * r
      v = .5 * r * r - aeth * (h1 + h2)
      w = aeth * r * h1
      epsr = 0.050
      rr = 2.0 * epsr
      jk = 1
c     WHILE ((jk .LT. 10).AND.(abs(rr) .GT. epsr))
c
c       Use Newton-Raphson iteration to solve Kerr's cubic equation
c       for reflection point range of the sea-reflected ray.  (This
c       equation may be solved explicitly using an inverse cosine.)
c       The Newton iteration works best if h1 is less than h2.
c
      DO WHILE ((jk .LT. 10).AND.(ABS(rr) .GT. epsr))
        jk = jk + 1
        r1sqrd = r1*r1
c         Kerr's cubic equation for reflection point range.
        fr1 = r1*r1sqrd + t*r1sqrd + v*r1 + w
c         Derivative of the cubic equation.
        fpr1 = 3.0*r1sqrd + 2.0*t*r1 + v
        rr = fr1/fpr1
        r1 = r1 - rr
        IF ((r1 .LT. 0.0).OR.(r1 .GT. r)) r1 = r/2.0
c     WEND
      END DO
      r2 = r - r1
      htp = h1 - r1*r1/ae2
      hrp = h2 - r2*r2/ae2
      psi = 1.0e-3 * htp / r1
      IF (psi .GT. 0.3) psi = ATAN(1.0e-3 * htp / r1)
      call ref(plr,psi,RMAG,PHI)
      pd = thefac*htp*hrp/r
      theta = pd + phi
      IF (ht .GE. hr) THEN
        gamma = r2/ae
      ELSE
        gamma = r1/ae
      END IF
      call opffac(gamma,r,psi,r1,r2,rmag,ALPHA,PATD,DR)
      fsqrd = patd*patd + dr*dr + 2.0*dr*patd*COS(theta)
c       Limit F-factor to -70 dB.
      IF (fsqrd .LT. 1.0e-7) fsqrd = 1.0e-7
      ff = - 10.0 * ALOG10(fsqrd)
      RETURN
      END

c
c  Subroutine PRNRNG
c
c  PRNTGT is used to print the surface search range table product.
```

```
c
c VARIABLE:       DESCRIPTION:
c   rng(3,5)    Surface search range table array of expected detection
c               ranges for the specified radars versus the 5 targets
c               and the minimum, average and maximum expected detection
c               ranges for each target.
c
      SUBROUTINE prnrng
c
      include 'surf.common'
c
      character*6 class(5)
      data class/"CV/CVN","CG/CGN","DD/DDG","FF/FFG","OSA II"/
c
      write(*,'("                          DETECTION RANGE (NMI)")')
      write(*,'("   SHIP TYPE/CLASS        MIN    AVERAGE   MAX")')
      do i = 1, 5
c       IF(i .EQ. 1) write(*,'("      CV/CVN ")')
c       IF(i .EQ. 2) write(*,'("      CG/CGN ")')
c       IF(i .EQ. 3) write(*,'("      DD/DDG ")')
c       IF(i .EQ. 4) write(*,'("      FF/FFG ")')
c       IF(i .EQ. 5) write(*,'("      OSA II ")')
        write(*,1000) class(i), rng(1,i), rng(2,i), rng(3,i)
      end do
 1000 format(8x,A6,11x,3(2x,f6.1))
      return
      end

c
c Subroutine PUSH
c
c PUSH stores elements in an array and returns.
c
c Variable:       Description:
c
c   array       iq array to hold data elements
c   iq          Size of data array
c   nq          Number of data elements stored in data array
c   data        The data element to be stored
c
c
      SUBROUTINE push(ARRAY,iq,nq,data)
c
      real*4 data,array
      integer*2 iq,nq
      dimension array(iq)
c
c       Shift array elements down one
c     do i=iq,2,-1
      DO j=2,iq
```

```
            i=iq-(j-2)
            array(i)=array(i-1)
         END DO
c
c          Store new data element in top of array
         array(1)=data
         nq=nq+1
         IF(nq .GT. iq) nq = iq
         RETURN
         END

c
c  Subroutine R1ITER
c
c  R1ITER determines a reflection point range 'r1' corresponding
c  to 'rtheta'.  The desired reflection point range is determined by
c  a Newton-Raphson iteration technique to vary the reflection point
c  point range until the correct value is found.
c
c  Variable:        Description:
c    r             Distance, or range, in km.
c    r1            Distance from the transmitting antenna to reflection
c                    point in km.
c    r2            Distance from the target/receiver antenna to the
c                    reflection point in km.
c    f             Function (Total path difference between direct and
c                    sea-reflected rays: Theta) used in iteration loop.
c    f1            Finite derivative of f.
c    icount        Iteration loop counter.
c    phi           Phase-lag due to sea-surface reflection - radians.
c    plr           EM system polarization [H = horizontal, V = vertical,
c                    C = circular].
c    psi           Grazing angle in radians.
c    r             Range, in km.
c    r1            Distance from the transmitting antenna to reflection
c                    point in km.
c    r2            Distance from the target/receiver antenna to the
c                    reflection point in km.
c    rmag          Magnitude of the reflection coefficient.
c    rtheta        The desired value of theta.
c
c
         SUBROUTINE r1iter(plr,rtheta,R1,R2,R,PSI,RMAG)
c
c
         real*4  f, f1, phi, psi, r, r1, r2, rmag, rr, rtheta
         character*1 plr
         integer*2 icount
c
         include 'ffac.common'
```

```
      include 'envsys.common'
c
      icount = 0
      rr = r1
c
c     WHILE ((abs(rr) .GT. 0.001).AND.(icount .LT. 100))
c         Equivalent to:  100 IF((...).and.(...))THEN
c                                 .....
c                                 GOTO 100
c
      DO WHILE ((abs(rr) .GT. 0.001).AND.(icount .LT. 100))
c
c         Calculate phase difference, theta, corresponding to
c         reflection point range r1.  Then use finite derivative
c         method to iterate to the range where theta is equal to
c         the target value: rtheta.
c
        call gtheta(plr,r1,R,F,R2,PSI,RMAG)
        call gtheta(plr,r1+0.001,R,F1,R2,PSI,RMAG)
        fp = (f1 - f) / 0.001
        rr = (rtheta - f) / fp
        icount = icount + 1
        IF (rr .GT. -r1) THEN
          IF (rr + r1 .LE. horzn1) THEN
             r1 = r1 + rr
          ELSE
             r1 = (r1+horzn1)/2.0
          END IF
        ELSE
          r1 = r1/2.0
        END IF
      END DO
c     WEND
      RETURN
      END
c

c
c  Subroutine RARRAY
c
c    RARRAY is used to determine several ranges to be used by the
c    LOOP subroutine.
c
c
      subroutine rarray(h50, r50)
c
      include 'surf.common'
      include 'envsys.common'
      include 'ffac.common'
c
```

```
      real*4  h50, pi, r50(3), twopi
c
      hr = h50
      hdif = (hr - ht) * 1.0e-3
      IF (hr .LT. ht) THEN
        h1 = hr
        h2 = ht
      ELSE
        h1 = ht
        h2 = hr
      END IF
      r50(1) = 370.00
      rsubd = rsdfac + 3.572*(SQRT(rkmin * hr) + SQRT(rkmin * ht))
      r50(2) = rsubd
      call oplimit(OPMAXD, OPMAXF)
      r50(3) = opmaxd
      return
      end

c
c  Subroutine REF
c
c  Subroutine REF returns the magnitude and phase lag of the reflection
c  coefficient for reflection from the (smooth) sea surface. These
c  quantities are calculated as a function of the grazing angle psi.
c  The complex square roots are done by separating the complex variables
c  into their real and imaginary parts.  No complex function calls are
c  used.
c
c
c  VARIABLE:        DESCRIPTION:
c    rnreal       Real part of the square of the index of refraction,
c                   (the dielectric constant of sea-water).
c    rnimag       Imaginary part of the square of the index of
c                   refraction (the conductivity of sea water
c                   times the wavelength times other constants).
c    phi          Phase change (lag) in radians.
c    plr          EM system antenna polarization: H = horizontal;
c                   V = vertical; C = Circular.
c    psi          Grazing angle in radians.
c    rmag         Magnitude of the reflection coefficient.
c    sinpsi       SIN(psi).
c
c    various      All variables not listed above are temporary.
c
      SUBROUTINE ref(plr, psi, RMAG, PHI)
c
      real*4 angrt, at, bt, ct, dt, phi, phiv, pi, psi,
     1       rcv, rmag, rmagrt, rtimag, rtreal,
     2       rvimag, rvreal, rx, sinpsi, x, y
```

```
      character*1 plr
c
      include 'ffac.common'
      include 'envsys.common'
c
      PI = 3.14159
c       Define RMAG, PHI for horizontal polarization.
      rmag = 1.0
      phi = PI
      IF (plr .NE. "H") THEN
c            Calculate RMAG, PHI for vertical polarization.
        sinpsi = SIN(psi)
        Y = rnimag
        X = rnreal - COS(psi)**2
        rmagrt = (x*x + y*y) ** 0.25
        angrt = ATAN(y/x) / 2.0
        rtreal = rmagrt * COS(angrt)
        rtimag = rmagrt * SIN(angrt)
        at = rnreal * sinpsi - rtreal
        ct = rnreal * sinpsi + rtreal
        bt = rnimag * sinpsi - rtimag
        dt = rnimag * sinpsi + rtimag
        rvreal = (at*ct + bt*dt) / (ct**2 + dt**2)
        rvimag = (bt*ct - at*dt) / (ct**2 + dt**2)
        rcv = SQRT(rvreal**2 + rvimag**2)
        IF (rvreal .NE. 0.0) THEN
          phiv = ATAN(rvimag/rvreal)
          IF (rvreal .LT. 0.0)  phiv = phiv + PI
        ELSE
          IF (rvimag .LT. 0.0)  phiv = -PI / 2.0
          IF (rvimag .GT. 0.0)  phiv = PI / 2.0
          IF (rvimag .EQ. 0.0)  phiv = 0.0
        END IF
        phiv = -phiv
        IF (phiv .LT. 0.0)  phiv = phiv + 2.0*PI
        rmag = rcv
        phi = phiv
        IF (plr .EQ. "C")THEN
c             Calculate RMAG, PHI for circular polarization.
          rx = SQRT(1.0 + rcv**2 + 2.0*rcv * COS(PI - phiv))
          rmag = rx/2.0
          a = rcv * SIN(phiv + PI) / rx
          a = ATAN( a/SQRT(1 - a*a) )
          phi = PI - a
          phi = -phi
          IF (phi .LT. 0.0) phi = phi + 2.0*PI
        END IF
       END IF
```

```
      RETURN
      END
c
c  Subroutine RTLOOP
c
c  LOOP returns the ranges where the reflected power from the radar
c  target is equal to the detection threshold for the radar system.
c  The detection ranges vary as a function of aspect angle and the 3
c  threshold values represent the minimum, average and maximum range
c  values where detection of the specific ship type would occur.
c
c  VARIABLE:        DESCRIPTION:
c    delr          Range difference, km.
c    fsterm        Free-space loss term, dB.
c    hgt(*)        Array containing the height intervals to be used
c                  by the FLOOP subroutine.
c    ht            Height of radar system antenna, m.
c    h50(i)        The height where approx. half of the radar cross-
c                  section is above and half below (m).
c    maxht         Maximum number of elements in the hgt(*) arrray.
c    plossr        Propagation loss at range r.
c    plr50(i)      Propagation loss for the ship target at range=r50(i).
c    r             Range iteration variable, km.
c    rend          Ending range for range loop, km.
c    rfs           Radar free-space range vs. a ship target, km.
c    rfs1          Radar free-space range vs. a 1 sq. meter target, km.
c    rfs1sm        Radar free-space range vs. a 1 sq. meter target, nmi.
c    rinst         Maximum instrumented range of the radar system, nmi.
c    rng(5,3)      Range array containg detection ranges for 5 targets
c                  and the three thresholds (km)
c    rstart        Start range for range loop, km.
c    r50(i)        Range array containing 370 km (200nmi), rsubd @ h50(i)
c                    (diffraction field minimum range) and the maximum
c                    range in the optical interference region at h50(i).
c    sigfac        Radar cross-section intermediate variable.
c    sigma         Radar cross-section in square-meters.
c    thresh        Radar detection threshold, dB.
c    th1           Radar detection threshold for smallest cross-section.
c    th2           Radar detection threshold for average cross-section.
c    th1           Radar detection threshold for largest cross-section.
c    wfi4          Weighted sum of all of the individual radiators of the
c                  ship target.
c
c
c
      SUBROUTINE rtloop
c
      include 'envsys.common'
      include 'ffac.common'
```

```
      include 'surf.common'
      real*4 delr, h50, plossr, plr50, rend, rfs, rfsl, rstart, r50
      real*4 sigfac, sigma, thresh, th1, th2, th3, wfi4
      integer*2 kn, maxht
      dimension  h50(5), r50(3), plr50(3)
      DATA h50/ 23.5, 15.5, 13.5, 11.5, 4.5 /
c
c       Initialize rng array to 0.0.
c
      Do i = 1,3
        DO m = 1,5
          rng(i,m) = 0.0
        END DO
      END DO
c
      rfsl = fsrlsm * 1.85
      sigfac = 52.0*SQRT(freq)
c
c  Target loop     (PROCESSING LOOP FOR THE TARGETS)
c
      DO  kn = 1, 5
c
c          Calculate target ship's radar cross section and free-space
c          range using ships displacement (ktons**3/2) and the radar
c          free space range for a 1 sq m target.  Calculate the path-
c          loss thresholds for the min/max/avg radar cross sections.
c
        sigma = sigfac * disp(kn)
        maxht = ielem(kn)
        rfs = rfsl * sigma**.25
        th2 = fsterm + 20.0*ALOG10(rfs)        ! THRESHOLD RCS AVG
        th3 = th2 + 13.0                       ! THRESHOLD RCS MAX
        th1 = th2 - 8.0                        ! THRESHOLD RCS MIN
c
        call rarray(h50(kn), R50)
c
        DO i = 1, 3
          r = r50(i)
          call floop(kn, maxht, r, WFI4)
          plr50(i) = fsterm + 20.0*ALOG10(r) - 5.0*ALOG10(wfi4)
        END DO
c
c     threshold loop:    radar threshold processing loop.
c
        DO m = 1, 3
          IF (m .EQ. 1) thresh = th1
          IF (m .EQ. 2) thresh = th2
          IF (m .EQ. 3) thresh = th3
c
```

```
c
          IF (thresh .GE. plr50(1)) THEN
c                Range is 200 nmi or max. instrumented range of radar.
              DO i = m, 3
                rng(i,kn) = AMIN1(rinst, 200.0)
              END DO
          ELSE
            rstart = r50(1)
            IF (thresh .GE. plr50(2)) THEN
c                Range is between 200 nmi and the diffraction region
c                minimum range for height h50(kn).
              rend = r50(2)
            ELSE
              IF (thresh .GE. plr50(3)) THEN
c                Range is between the diffraction region minimum range
c                and optical region maximum range for height h50(kn).
                rstart = r50(2)
                rend = r50(3)
              ELSE
c                 Target range is in the optical region.
                r = r50(3)
                plossr = plr50(3)
                DO WHILE ((r .GT. 5.0).AND.(thresh .LT. plossr))
                  r = r - 1.0
                  call floop(kn, maxht, r, WFI4)
                  plossr = fsterm + 20.0*ALOG10(r) - 5.0*ALOG10(wfi4)
                END DO
                IF (r .GT. 5.0) THEN
                  rstart = r + 1.0
                  rend = r
                ELSE
c                   Minimum range is 5 km (2.7) nmi.
                  rstart = 5.0
                  rend = 5.0
                END IF
              END IF
            END IF
            delr = (rstart - rend)/2.0
            r = rend + delr
c              Range Loop. - Loop used to determine the detection range
c              between two known ranges - rstart and rend.
            DO WHILE (delr .GT. 0.10)
              delr = delr/2.0
              call floop(kn, maxht, r, WFI4)
              plossr = fsterm + 20.0*ALOG10(r) - 5.0*ALOG10(wfi4)
              IF (thresh .GT. plossr) THEN
                r = r + delr
              ELSE
                r = r - delr
```

```
            END IF
          END DO
          rng(m,kn) = AMIN1(rinst, r/1.85)
        END IF
c
c       End threshold loop
c
       END DO
c
c     End target loop
c
      END DO
      RETURN
      END

c
c   Subroutine RUFF
c
c   Subroutine RUFF returns the sea-surface roughness correction for
c   the magnitude of the sea-reflected ray.
c
c
c   VARIABLE:        DESCRIPTION:
c     hbar           rms wave height in meters.
c     hbfreq         (2*PI*hbar)/wavelength.
c     hfpsi          (hbar*psi)/wavelength.
c     psi            Grazing angle in radians.
c     sinpsi         SIN(psi).
c     rufco          Sea-surface roughness coefficient.
c
c
      SUBROUTINE ruff(hbar, hbfreq, psi, sinpsi, RUFCO)
c
      real*4  hbar, hbfreq, hfpsi, psi, rufco, sinpsi
c
c
      rufco = 1.0
      IF (hbar .NE. 0.0) THEN
         hfpsi = hbfreq * psi * 0.159155
         IF (hfpsi .LE. 0.11) THEN
           rufco = EXP((-2.0) * (hbfreq*sinpsi)**2)
         ELSEIF (hfpsi .LE. 0.26) THEN
           rufco = 0.5018913 - SQRT(0.2090248 - (hfpsi - 0.55189)**2)
         ELSE
           rufco = 0.15
         END IF
      END IF
c
      RETURN
      END
```

```
c
c  Subroutine SYSFIL
c
c  SYSFIL list available system files and allows the user to select a
c  radar system file.
c
c  Variable:        Description:
c    antena       Height of radar system antenna in m.
c    antype       Antenna type:
c                   O = omnidirectional
c                   S = sin(x)/x
c                   C = cosecant-squared
c                   H = generic height-finder
c    bwidth       Antenna beam width in degrees.
c    elevat       Antenna elevation angle in degrees.
c    filename     Name of System file.
c    freq         Radar system frequency in MHz.
c    fsrng        Radar system free-space range vs. 1 sq. meter target.
c    polar        Antenna polarization:
c                   H = horizontal
c                   V = vertical
c                   C = circular.
c    rinst        Maximum instrumented range of radar system (nmi).
c
c
      SUBROUTINE sysfil(freq, antena, fsrng, rinst, polar, antype,
     1                  bwidth, elevat)
c
c
      real*4 antena, bwidth, elevat, freq, fsrng, rinst
      integer*2 ZR, ZW
      character*1 antype, polar
      character*20 filename
c
c       Initialize read and write channels.
      ZR = 5
      ZW = 6
c
      call system ('ls [S]* 1>&2'//char(0))
      write (ZW,'(//,"Enter input file name: ",$)')
      read (ZR,'(a12)') filename
      open (10,FILE=filename)
c
      read (10, '(f10.1)') freq              ! radar freq
      read (10, '(f10.1)') antena            ! radar antenna ht
      read (10, '(f10.1)') fsrng             ! radar free-space range
      read (10, '(f10.1)') rinst             ! radar instrumented range
      read (10, '(a1)') polar                ! antenna polarization
      read (10, '(a1)') antype               ! antenna type
```

```
      read (10, '(f10.1)') bwidth            ! vert. beam width
      read (10, '(f10.1)') elevat            ! ant. elev. angle
      close(10)
c
      RETURN
      END

c
c  Subroutine SYSINP
c
c  Subroutine SYSINP prompts the user for EM system parameters and re-
c  turns.  System parameters can be entered from the keyboard or from a
c  file.  If the system is entered from the keyboard it can be stored in
c  a file.
c
c  Variable:        Description:
c    atype         Antenna type:
c                    O = omnidirectional
c                    S = sin(x)/x
c                    C = cosecant-squared
c                    H = height-finder
c    beam          Beam width in degrees.
c    elang         Antenna pointing (elevation) angle in degrees.
c    fmhz          EM system frequency in MHz.
c    fsrlsm        Radar free-space range vs. a 1 sq. meter target (nmi).
c    filename      EM system filename.
c    plr           Antenna polarization:
c                    H = horizontal
c                    V = vertical
c                    C = circular
c    rinst         Maximum instrumented range of the radar system (nmi).
c    xmtr          EM system antenna height, m.
c
c
      SUBROUTINE sysinp(FMHZ, XMTR, FSR1SM, RINST, PLR, ATYPE,
     1                  BEAM, ELANG)
c
c
      real*4 beam, elang, fmhz, fsrlsm, rinst, xmtr
      character*1 atype, dummy, plr
      character*20 filename
      integer*2 ZW, ZR
c
c       Specify the read (5) and write (6) channel numbers.
      ZW = 6
      ZR = 5
c
c       Enter EM system parameters from file or keyboard.
      write(ZW,'("You may enter EM system from a file or keyboard.")')
      write(ZW,'("Enter EM system data from file? (yes or no) ",$)')
```

```
      read(zr,'(A1)')dummy
      IF ((dummy(1:1) .eq. 'y').or.(dummy(1:1) .eq. 'Y')) THEN
c       Enter EM system data from a file.
        call sysfil(FMHZ,XMTR,FSR1SM,RINST,PLR,ATYPE,BEAM,ELANG)
      ELSE
c       Enter the EM system parameters from keyboard.
        write(ZW,'("Enter Radar System Parameters: ")')
c
c         Initialize EM system variables.
        fmhz = 5600.0
        xmtr = 25.0
        fsr1sm = 25.0
        rinst = 200.0
        plr = "H"
        atype = "O"
        beam = 0.0
        elang = 0.0
c
        write(ZW,1000)
 1000   format('Enter radar frequency in MHz (100 to 20,000) ',$)
        read(ZR,*) fmhz
        IF (fmhz .LT. 100.0) fmhz = 100.0
        IF (fmhz .GT. 20000.0) fmhz = 20000.0
c
        write(ZW,1100)
 1100   format('Enter radar transmitter height in meters (1 to 100) ',$)
        read(ZR,*) xmtr
        IF (xmtr .LT. 1.0) xmtr = 1.0
        IF (xmtr .GT. 100.0) xmtr = 100.0
c
        write(ZW,1200)
 1200 format('Enter radar free-space range vs. 1 sq.meter target'
     1              ' (1 to 1000)nmi ',$)
        read(ZR,*) fsr1sm
        IF (fsr1sm .LT. 1.0) fsr1sm = 1.0
        IF (fsr1sm .GT. 1000.0) fsr1sm = 1000.0
c
        write(ZW,1300)
 1300 format('Enter radar maximum instrumented range (10 to 200)nmi ',$)
        read(ZR,*) rinst
        IF (fsr1sm .LT. 10.0) rinst = 10.0
        IF (fsr1sm .GT. 200.0) rinst = 200.0
c
        write(ZW,1400)
 1400   format('Enter radar system polarization (H, V, C) ',$)
        read(ZR,'(A1)') plr
        IF ((plr .EQ. "c") .OR. (plr .EQ. "C")) plr = "C"
        IF ((plr .EQ. "v") .OR. (plr .EQ. "V")) plr = "V"
        IF ((plr .NE. "V") .AND. (plr .NE. "C")) plr = "H"
```

```
c
      write(ZW,1500)
 1500 format('Enter antenna type - options are: Omnidirectional, '
    1 /,'Sin(x)/x, Cosecant-squared, Height-finder (O, S, C, H)) ',$)
      read(ZR,'(A1)') dummy
      IF ((dummy .EQ. "o") .OR. (dummy .EQ. "O")) atype = "O"
      IF ((dummy .EQ. "s") .OR. (dummy .EQ. "S")) atype = "S"
      IF ((dummy .EQ. "c") .OR. (dummy .EQ. "C")) atype = "C"
      IF ((dummy .EQ. "h") .OR. (dummy .EQ. "H")) atype = "H"
      IF ((atype .NE. "S") .AND. (atype .NE. "H") .AND.
    1     (atype .NE. "C")) atype = "O"
      beam = 0.0
      elang = 0.0
      IF(atype .NE. "O") THEN
        write(ZW,1510)
 1510   format('Enter antenna beam width in degrees (>0.0 to 45) ',$)
        read(ZR,*) beam
        IF (beam .LE. 0.0) beam = 0.10
        IF (beam .GT. 45.0) beam = 45.0
      write(ZW,1520)
 1520 format('Enter antenna elevation angle in degrees (-10.0 to 10.0)'
    1      /,'(0 is normal) ',$)
        read(ZR,*) elang
        IF (elang .LT. -10.0) elang = -10.0
        IF (elang .GT. 10.0) elang = 10.0
      END IF
      write(ZW,'("Do you wish to store this EM system in a file?",
    1  " (yes or no) ",$)')
      read(zr,'(A)')dummy
      IF ((dummy(1:1) .eq. 'y').or.(dummy(1:1) .eq. 'Y')) THEN
        write (ZW,'(" Current System Files: ")')
        call system ('ls [S]* 1>&2'//char(0))
        write (ZW, 1600)
 1600   format("Enter file name (First letter MUST be S): ",$)
        read (ZR,'(a12)') filename
        open (10,FILE=filename)
c     Write frequency, antenna height, free-space range and
c     maximum instrumented range in file.
        write(10, '(f10.1)') fmhz
        write(10, '(f10.1)') xmtr
        write(10, '(f10.1)') fsrlsm
        write(10, '(f10.1)') rinst
c     Write the antenna characteristics in file.
        write(10, '(a1)') plr
        write(10, '(a1)') atype
        write(10, '(f10.1)') beam
        write(10, '(f10.1)') elang
c         close file
        close(10)
```

```
        END IF
      END IF
c
      RETURN
      END

c
c  Subroutine TROPO
c
c  Subroutine TROPO returns the troposcatter loss for a given range.
c  Troposcatter loss is based on models by Yeh with a frequency-gain
c  correction term, HO, from National Bureau of Standards Document
c  NBS 101.  Frequency gain factor gives additional loss for low
c  frequency, low-sited antennas.
c
c  VARIABLE:        DESCRIPTION:
c    ae            Effective earth radius in kilometers
c    exloss        Antenna gain for lowest optical region ray in dB
c    horizn        Horizon range in kilometers
c    h1            Transmitter height in meters
c    h2            Radar target/receiver height in meters
c    h0            Frequency gain factor in dB
c    r             Ground range in km
c    rnsterm       Constant involving Surface Modified refractivity
c    rnsubs        Modified refractivity value at the sea surface
c    rone          4*PI*h1*ttot/wavelength
c    rtwo          4*PI*h2*ttot/wavelength
c    tfac          Troposcatter region constant
c    tloss         Troposcatter loss in dB
c    tsub0         Angle, theta sub 0, associated with total range r
c    tsub1         Angle, theta sub 1, associated with horizon range r1
c    tsub2         Angle, theta sub 2, associated with horizon range r2
c    ttot          Scattering angle, (theta in NBS 101)
c
c    various       All variables not listed above are temporary.  Most
c                    variables use the names given in NBS 101.
c
      SUBROUTINE tropo(r,tloss)
c
c
      real*4 chi, csub1, csub2, delh0, etas, horizn, hsub0, h0,
     1       h0r1, h0r2, q, r, rnsterm, rnsubs, rone, rtwo,
     2       s, tfac, tloss, tsub0, tsub1, tsub2, ttot, zeta
c
      include 'ffac.common'
      include 'envsys.common'
c
      rnsubs = Munits(1)
      tfac = 0.08984/rk
      horizn = 3.572 * ( SQRT(rk*h1) + SQRT(rk*h2) )
```

```
rnsterm = 0.031 - 0.00232 * rnsubs + 5.67E-6 * rnsubs*rnsubs
tsub0 = r / ae
tsub1 = SQRT(h1 * ae/500.0) / ae
tsub2 = SQRT(h2 * ae/500.0) / ae
ttot = tsub0 - tsub1 - tsub2
zeta = ttot/2.0 + tsub1 + (h1 - h2) / (1000.0*r)
chi  = ttot/2.0 + tsub2 + (h2 - h1) / (1000.0*r)
rone = h1 * 0.0419 * freq * ttot
rtwo = h2 * 0.0419 * freq * ttot
IF (rone .LT. 0.1) rone = 0.1
IF (rtwo .LT. 0.1) rtwo = 0.1
s = zeta / chi
IF (s .GT. 10.0) s = 10.0
IF (s .LT. 0.1)  s = 0.1
q = rtwo / (s * rone)
IF (q .GT. 10.0) q = 10.0
IF (q .LT. 0.1)  q = 0.1
hsub0 = s * r * ttot / (1.0 + s)**2
etas = 0.5696*hsub0 * (1.0 + rnsterm*EXP(-3.8e-6 * hsub0**6))
IF (etas .GT. 5.0)  etas = 5.0
IF (etas .LT. 0.01) etas = 0.01
csub1 = 16.3 + 13.3*etas
csub2 = 0.4 + 0.16*etas
h0r1 = csub1 * (rone + csub2)**(-1.333)
h0r2 = csub1 * (rtwo + csub2)**(-1.333)
h0 = (h0r1 + h0r2) / 2.0
delh0 = 1.13 * (0.6 - ALOG10(etas)) * ALOG(s) * ALOG(q)
IF (delh0 .GT. h0) THEN
  h0 = 2.0*h0
ELSE
  h0 = h0 + delh0
END IF
IF (h0 .LT. 0.0) h0 = 0.0
tloss = 114.9 + tfac*(r-horizn) + 10.0*ALOG10(r*r*freq**3)
tloss = tloss - rnsubs*0.2 + h0 + exloss
RETURN
END
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1990	Final: Jan 1990–Jun 1990

4. TITLE AND SUBTITLE

SPECIFICATION FOR A SURFACE-SEARCH RADAR-DETECTION-RANGE MODEL

5. FUNDING NUMBERS

6. AUTHOR(S)

C. P. Hattan

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Ocean Systems Center
San Diego, CA 92152-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

NOSC TD 1930

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Oceanographic Office
NSTL Station
Bay St. Louis, MS 39522

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A model that predicts surface-search radar detection range versus a variety of combatants has been developed at the Naval Ocean Systems Center. This model uses a simplified ship radar cross section (RCS) model and the U.S. Navy Oceanographic and Atmospheric Mission Library Standard Electromagnetic Propagation Model. It provides the user with a method of assessing the effects of the environment on the performance of a surface-search radar system. The software implementation of the model is written in ANSI Fortran 77, with MIL–STD–1753 extensions. The program provides the user with a table of expected detection ranges when the model is supplied with the proper environmental and radar system inputs. The target model includes the variation in RCS as a function of aspect angle and the distribution of reflected radar energy as a function of height above the waterline. The modeled propagation effects include refraction caused by a multisegmented refractivity profile, sea-surface roughness caused by local winds, evaporation ducting, and surface-based ducts caused by atmospheric layering.

14. SUBJECT TERMS

radar cross section
electromagnetic propagation

15. NUMBER OF PAGES

71

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAME AS REPORT